CONNEXIONS™

The Interoperability Report

January 1991 | *Special Issue: Inter-domain Routing* | Volume 5, No. 1

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop, Inc., 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779.

ISSN 0894–5926

From the Editor

This special issue of *ConneXions* deals with *inter-domain routing*, and related topics. Our previous special issue on routing, published in August 1989, focused on *intra-domain routing* issues and specifically avoided any mention of what happens when routing takes place between diverse network communities, operated by different administrations and often running different protocol suites. This companion issue will examine the current state of inter-domain routing in the Internet.

Our first article, written by Paul Tsuchiya of Bellcore, introduces the inter-domain routing concept. This is followed by a slightly more detailed look at inter-domain routing. The second article is written by Robert Woodburn of SAIC who has been involved in the Open Routing Working Group of the IETF.

Martha Steenstrup of BBN Communications Corporation discusses the *policy* aspects of inter-domain routing in an article starting on page 19. This is followed by an overview of the *Border Gateway Protocol* (BGP). BGP is the latest member of the "routing protocol family" in the Internet. It was designed as a replacement for the original Exterior Gateway Protocol (EGP2) which became unsuitable for use in the evolving and ever expanding Internet.

The changing face of the Internet has also led to some important changes in the rules that govern how organizations "join the club". These changes were stipulated in RFC 1174 and are summarized in an article by Daniel Dern on page 30.

A crucial aspect of all routing protocols is the nature of the *addresses*. To scale well, addresses must be hierarchical. However, hierarchical addresses impose constraints on the paths found by routing, thus perhaps getting in the way of policy. Researchers are currently struggling with these opposing needs. Other addressing problems exist. The number of available IP addresses may be running out. The procedures for NSAP address assignment in OSI are still in flux, and it cannot be assumed that NSAP addresses, as large as they are, will have the hierarchical structure needed for scaling. We hope to explore these topics in a future issue of *ConneXions*.

I am happy to announce the arrival of Rich Tennant's *The 5th Wave* cartoons in *ConneXions*. When I first saw the cartoon included in this month's issue (page 23), I simply could not resist. While I cannot promise these cartoons as a "regular" feature, they will appear from time to time. We hope you'll appreciate this lighter addition to an otherwise highly technical journal. Speaking of contents, we really do appreciate your suggestions for topics, so do write or call!

Inter-domain Routing in the Internet:

The People Power Revolution Continues

by Paul F. Tsuchiya, Bell Communications Research

Introduction

Nobody likes to be told what to do. This applies to how we run and use our networks as much as anything else. One of the great promises of internet technology (TCP/IP and similar technologies) is the freedom to network as we please. For instance, by putting the job of reliable packet delivery into the user's computer, the user is freed from the underlying network technology—any networking technology can be introduced as easily as any other. We are released from dependency on the "value-added" (connection-oriented) network.

However, transition to people power is not complete. In most cases, once a packet leaves our computer, and especially our private network, we have lost control over it. We cannot necessarily insure that our packets will traverse networks that offer an acceptable *Quality-of-Service* (QOS), or price. This is the job of *Inter-domain routing* (IDR), often associated with *policy routing*. Once IDR becomes reality, we will be able to choose where our packets go. Connection-by-connection, we will be able to pick among the available offered services (both inside and outside our domain), and shop for price.

I said once IDR becomes reality. Perhaps I should have said if or to what extent IDR becomes reality. No routing algorithm is trivial, and the functions required by and constraints placed on IDR make it particularly difficult. Over the past few years, considerable research effort has been devoted to IDR, and now there are two competing proposals. In this article, we describe and evaluate those two proposals, give their status, and speculate on what we might see in the future.

What is Inter-Domain Routing?

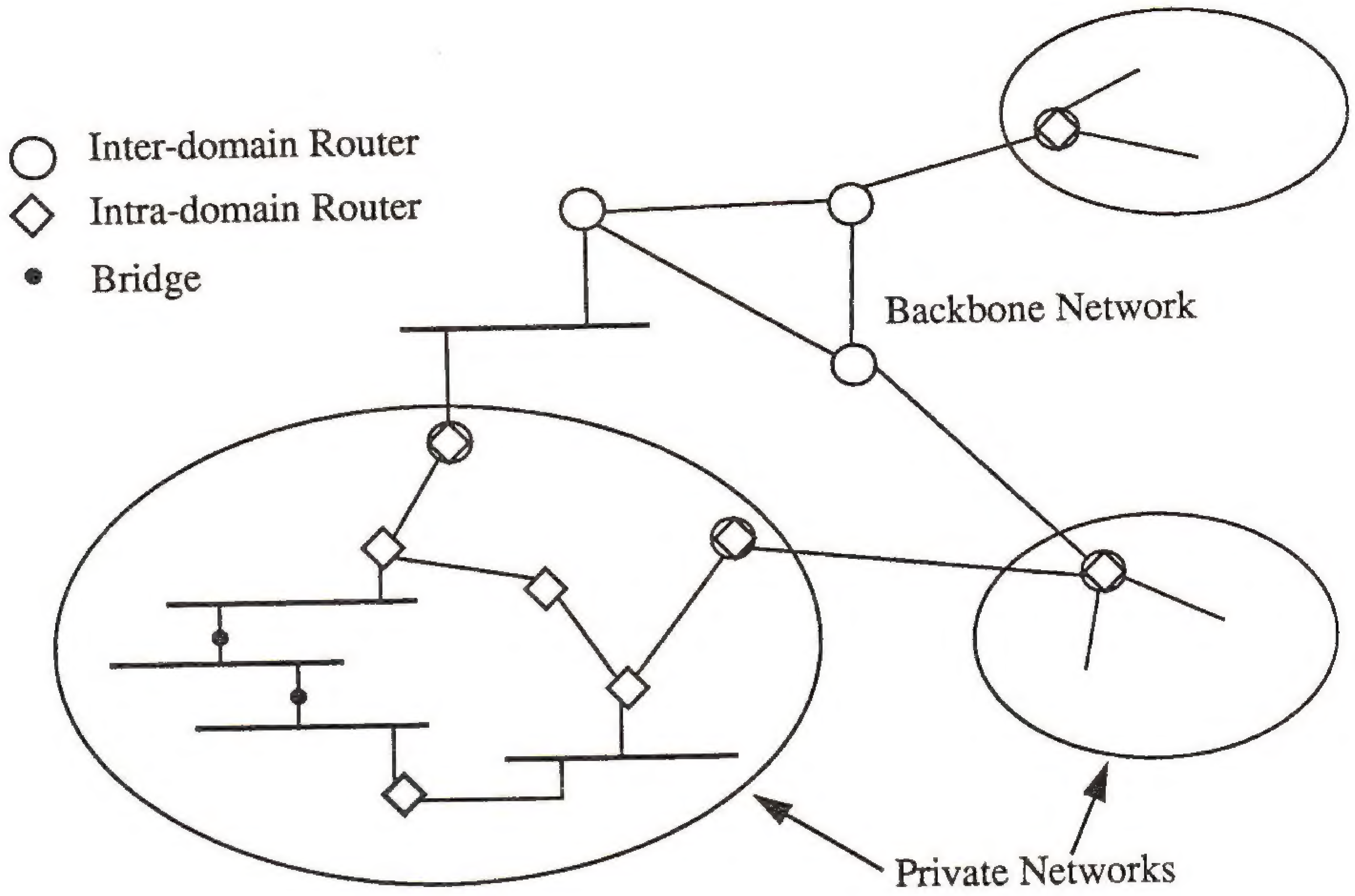
Internet routing can be partitioned into two parts: routing within a private network (called *intra-domain routing*), and routing between private networks. While these two influence each other, they are to the extent possible decoupled (see Figure 1). Within a private network, there is (one hopes) full trust between routers, and coordinated administration of routers. Therefore, the major requirement placed on intra-domain routing is performance-the routing algorithm should respond immediately to failures, should find the highest-bandwidth paths, and so on.

Existing standards

Standards for scalable, high-performance intra-domain routing exist for both IP and OSI 8473 (the OSI equivalent to IP). For OSI 8473, it is the IS-IS (intermediate-system to intermediate-system) routing protocol, ISO 10589. For IP, it is both OSPF (*Open Shortest Path First*) and a dual protocol version of IS-IS. While these intra-domain routing protocols accept information from IDRs, they are largely decoupled from the IDRs. This article does not concern itself with intra-domain routing protocols.

However, between domains there is little coordinated administration, and therefore some level of mistrust. For instance, one cannot prevent failures from occurring in other private networks or in backbones, and so firewalls must exist for preventing the effects of such failures from spreading into other private networks. Also, there is a greater need for policy in IDR. One wants to avoid backbones that will not accept one's packets, and traverse backbones that offer an appropriate QOS or price.

Finally, IDR must scale well. The size of the global Internet is essentially unbounded. The mechanisms for efficiently dealing with that size are mandatory. Therefore, the major requirements placed on IDR are for firewalls, scaling and policy. Performance, while still important, is secondary.



In the Internet, there are at least three layers of routing. Bridges route among each other across LANs. They use the LAN (or MAC) address. Intra-domain routers route amongst each other within a private network. They use the intra-domain part of the internet (IP or ISO 8473) address. Inter-domain routers route among each other between private networks, usually including backbone networks. They look at the inter-domain part of the internet address. Bridges and routers do not share information. Intra- and Inter-domain routers do share information, often by virtue of a single router taking on both roles.

Figure 1: Layers Upon Layers of Routing

Where are we now?

One might ask, "You say we need all this IDR to connect our private networks, and yet our private networks seem to be quite connected already, so what is it we have now?" To answer, let me first take a step back and get some perspective.

Currently, there is a plethora of networks and internetworks, private and otherwise (for instance, TCP/IP, OSI, SNA, DECnet, BITNET, UUCP, X.25, AppleTalk, and more). By and large, to the extent users of these different networks and network technologies can communicate, it is through (application layer) electronic mail gateways. This, in fact, involves a level of routing above those shown in Figure 1 (for instance, between mail systems).

Of these, this article focuses only on the internet joined by IP, and in the future, OSI 8473 routers. This internet (which we call simply the Internet with a capital "I") alone consists of over 2000 TCP/ IP-based private networks, which are connected by regional, national, and international backbones utilizing both IP and X.25 switching. The Internet is nearly doubling in user population each year.

continued on next page

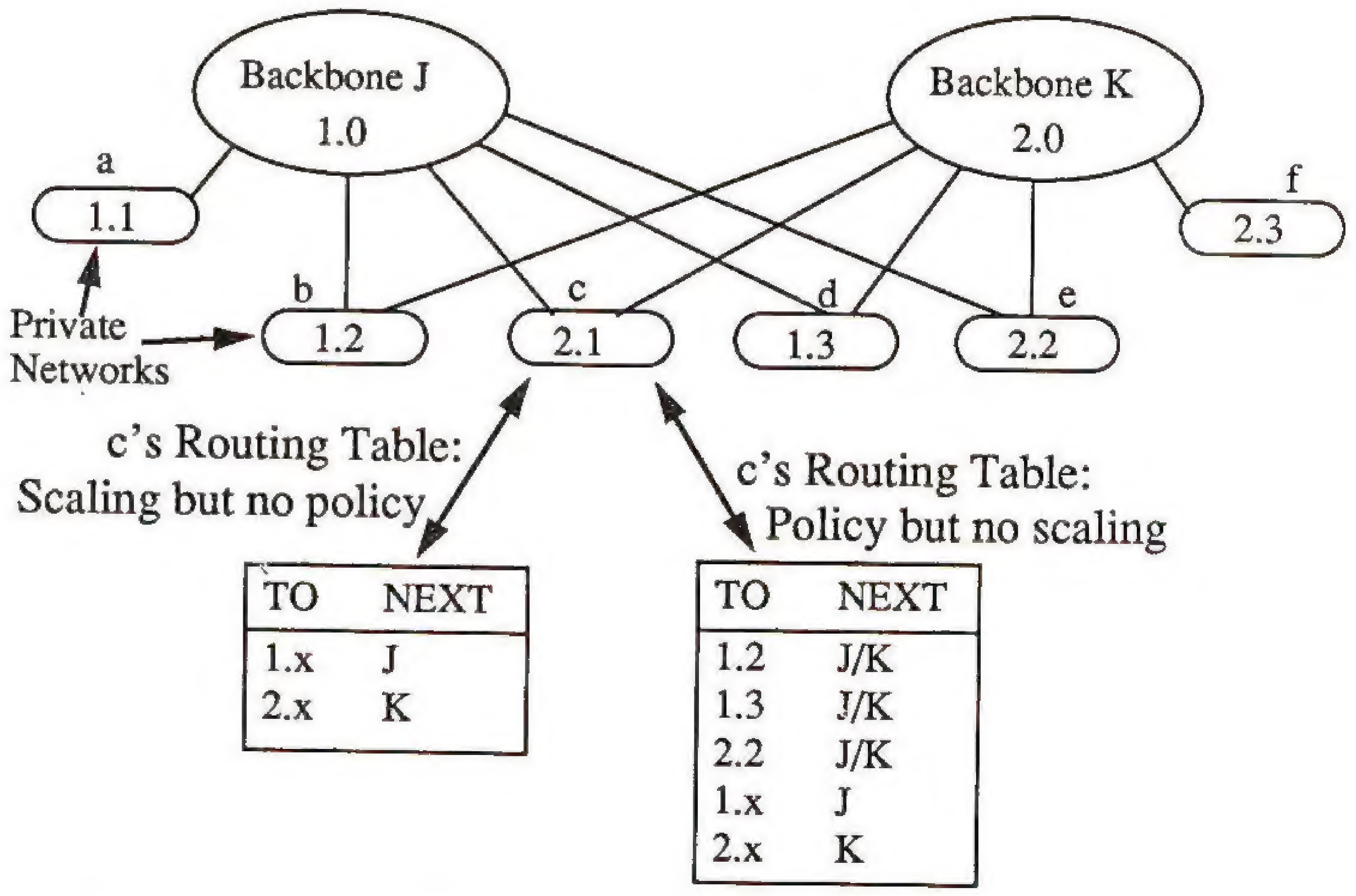
Inter-domain Routing in the Internet *(continued)*

Currently, the Internet is held together by a patchwork of static routing table entries and routing protocols with names such as EGP, RIP, and *gated,* and by the network wizards (of which I am not one) who administer the whole thing. The network wizards determine which routers should talk to which other routers, and about whom, in such a way that loops (where packets return to a router they have already visited) are avoided and that the resulting paths are pretty much what people expect. However, loops often occur, usually because someone connects up a router and starts exchanging routing information without first consulting one of the wizards. In addition, this patchwork of routing protocols does not scale well, and so many routers must periodically be fitted with more memory to handle the internet growth.

Unless, I suppose, you are a wizard interested in job security, this is not a satisfactory state of affairs. Indeed, solving the IDR problem is one of the top agenda items of both the Internet Engineering Task Force (IETF), the standards and engineering body that maintains the TCP/IP suite of protocols, and of ANSI X3S3.3, the USA standards body maintaining the OSI equivalents to TCP/IP (ISO 8073 Class 4 and ISO 8473 respectively).

Why is Inter-Domain Routing so hard?

One of the fundamental reasons why IDR is hard is because it tries to satisfy contradictory goals: scaling and policy (see Figure 2). The most effective way to achieve scaling is through hierarchical addressing. With hierarchical addressing, a router only needs summary information about things far away. For an example using a hierarchical address familiar to all of us, a telephone switch on the west coast of the USA doesn't care about the individual telephone numbers or necessarily even office codes for switches on the east coast—it usually only concerns itself with the relatively few area codes. (At the time of this writing there were only 137 active area codes in the Integrated North American Numbering Plan Area).



TO	NEXT
1.x	J
2.x	K

TO	NEXT
1.2	J/K
1.3	J/K
2.2	J/K
1.x	J
2.x	K

Given the topology and addressing shown, the IDR router in private network c can keep entries for only the backbones J and K, in which case it has a small routing table but can't find multiple paths to private networks b, d, and e (scaling but no policy). Or, c can additionally keep explicit entries for b, d, and e, in which case c can find multiple paths, but has a large routing table (policy but no scaling).

Figure 2: Why Scaling and Policy don't mix well

However, if one wants to find multiple paths to private networks far away, say through different backbone networks, and if not all private networks are reachable through all backbones (unlike the case with the telephone system, where everybody can be reached through AT&T, MCI, and Sprint), then simple hierarchical addresses don't work. The very information we need to know (which private networks are connected to which backbones, for instance) is hidden by the hierarchical addresses.

Proposed solutions

There are two solutions to IDR on the table. Unfortunately, their names are almost identical-IDPR and IDRP, for *Inter-Domain Policy Routing* and *Inter-Domain Routing Protocol* respectively. To make reading less confusing, we shall spell *IDRP* in italics.

Its appropriate to give the history of these protocols, as it's hard to fathom why things are the way they are without it. Several years ago, when it became critical that a solution to IDR be found, a Working Group within the IETF, called the *Open Routing Working Group* (ORWG), was formed. While this group initially did not have policy as one of its requirements, it soon became apparent that policy was indeed an important requirement. This change of focus was influenced heavily by the *Autonomous Networks Working Group* in the IRTF (Internet Research Task Force, a sister group of the IETF, but interested more in research), which was looking at issues of all kinds that arise between separately administered networks, including, for instance, security.

BGP

In the mean time, the routing problem in the Internet was reaching crisis proportion, and so another group, not wanting to wait for a solution to come out of the ORWG, designed a very simple IDR protocol called the *Border Gateway Protocol* (BGP). While BGP has limited policy and virtually no scaling, it is head and shoulders better than the existing patchwork, mainly by virtue of the fact that it detects and prevents loops. Test implementations of BGP were developed in a few weeks, after which additional engineering was done in another IETF Working Group, the *Interconnectivity Working Group* (IWG), resulting in an improved version of BGP (BGP2). [See article on page 24].

As of this writing, BGP2 is approved as a Proposed Internet Standard, and is documented in RFCs 1163 and 1164 (Request For Comments, the series of documents that contain, among other things, standardized TCP/IP suite protocols). In addition, several implementations of BGP2 exist, for instance in NSFNet backbone routers, cisco routers, and UNIX. BGP2 is currently being deployed, and will soon replace large parts of the existing patchwork.

At about the same time that the ORWG started its work, work was also underway in ISO to design an IDR protocol (both in X3S3.3 and in ECMA, the *European Computer Manufacturers Association*). The work from ECMA eventually was proposed as a base text in ISO, but did not have wide support among the USA delegation. However, the inventors of BGP brought it to X3S3.3, where it was found to be preferable to the ECMA work. Additional work took place in X3S3.3 on BGP, where it is called *IDRP*, and the USA has proposed that *IDRP* replace the ECMA work as the base text in ISO. This appears very likely to happen. Work on *IDRP* is ongoing in X3S3.3, where significant improvements in its scaling and policy capabilities are being made. Test implementations of a base set of *IDRP* features are currently being planned but are not yet underway. A protocol has been designed by the ORWG, called IDPR. Test implementations of IDPR are currently underway.

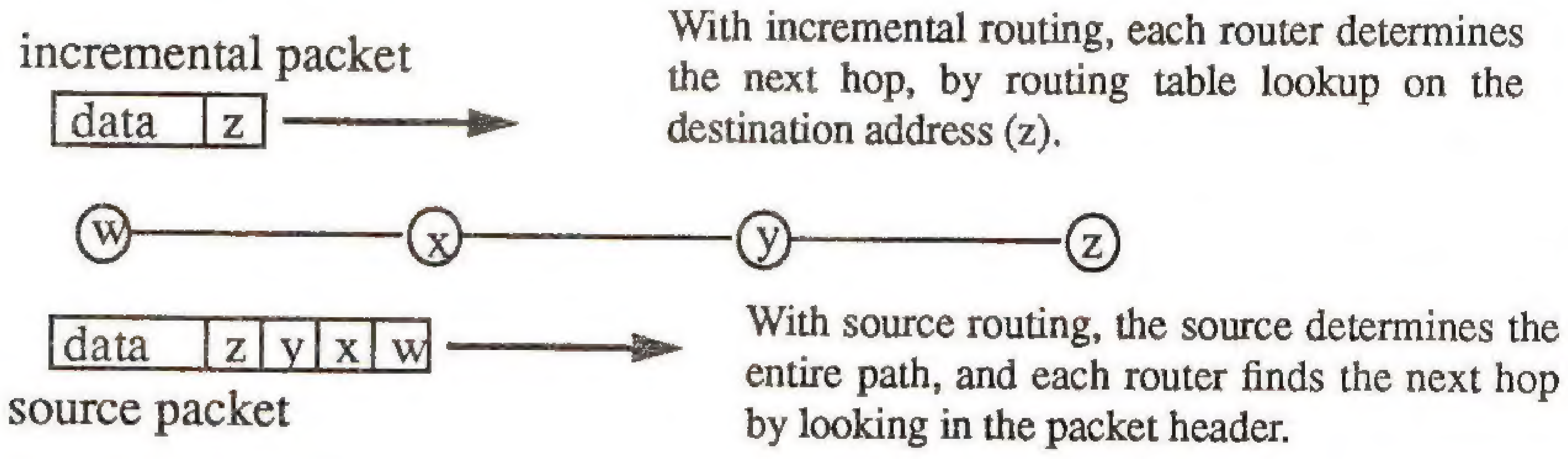
continued on next page

Inter-domain Routing in the Internet *(continued)*

IDRP

Now we describe the two protocols. We start with *IDRP,* as it is closer in design to most current routing protocols. *IDRP* may be classified as incremental, distance-vector routing. *Incremental routing* is the technique whereby each router makes the decision as to the next hop on the path to the destination (Figure 3). (This is contrasted with *source routing,* where the source of the packet determines the entire path, and encodes the path in each packet, or in a packet setup.)

Distance-vector is one of two common distributed algorithms for calculating routing tables. With distance-vector, each node tells each other node how far it thinks it is from the destination. After a number of such exchanges, the shortest path is found. For instance, in Figure 3, y would tell x it is one hop from z, and x would tell w it was two hops from z. Assume that w had some other neighbor e that said it was three hops from z. Based on the received information, w would know that x was the best way to get to z. *IDRP* has solved the looping problems typically associated with distance-vector algorithms by keeping the entire path to each destination in the routing table (although not in each packet).



With incremental routing, each router determines the next hop, by routing table lookup on the destination address (z).

With source routing, the source determines the entire path, and each router finds the next hop by looking in the packet header.

Figure 3: Incremental versus source routing

To achieve scaling, *IDRP* relies on hierarchical addresses (recall the phone number example). However, *IDRP* does not require that the hierarchical structure of the address be known in advance (like they are with phone numbers). IDRP associates a bit mask with each address in the routing table. The 1's in the mask determine which part of the hierarchical address is meaningful, and the 0's in the mask indicate which part of the address can be ignored. This way, any hierarchical address structure is acceptable, and different addresses may have different structure.

IDRP uses two techniques to achieve policy. First, *IDRP* associates attributes with each path to a destination, one for each policy. For instance, *IDRP* might calculate two paths to a destination, one for cost and one for bandwidth. Consider Figure 2. Assume that the path through J is high speed but expensive, and the path through K is low speed but cheap. J would then tell c, for instance, that it has a 45Mbps path for $10. K would tell see it has a 1.5Mbps path for $1. A user could now choose a low cost or a high bandwidth path, depending on its policy for a particular connection. The user marks the QOS desired for a packet using the QOS option in ISO 8473.

The cost of this technique is rather high, and so it should be used sparingly (it relates to the "policy, but no scaling" version of c's routing table). For each destination, the routers must keep a separate path per policy. For instance, if a destination has three policies associated with it, the routing algorithm must maintain three paths-the overhead is the same as if there were three separate destinations.

The second technique for getting policy is as effective as the first, but has much less overhead. Instead of keeping multiple paths to destinations, each destination is assigned multiple addresses, one per policy. For instance, consider again Figure 2. Notice that the dual-homed private networks picked either address 1.x or 2.x. However, if each dual-homed private network had two addresses (1.x and 2.x), then, depending on whether the source picked an address starting with 1 or with 2, the packets would travel either through backbone J or K. On one hand, policy is achieved by picking the appropriate address, but on the other, the routing table still need only keep track of the backbones, not each private network. This way, one gets both policy and scaling.

IDPR

IDPR may be classified as source, *link-state* routing. Most existing routing algorithms are link-state (for instance, IS-IS and OSPF), but very few use source routing. Like distance-vector, link-state is distributed. However, in link-state the raw topology information is flooded to each node, which then calculates paths by running a spanning tree algorithm on the collected topology graph.

Source routing gives IDPR some nice features. With *IDRP*, backbones must know the policies of the private networks. For instance, if the private network wants to be able to choose between price and bandwidth, the backbones must know this and pass on the appropriate information. This is fine with common policies such as price and bandwidth, but if the private network has an unusual policy constraint (for instance, cannot use a backbone subsidized by the government for non-research transmissions), then the backbones must pass on information accordingly. Since the cost of QOS routing in *IDRP* is relatively high, this effectively prohibits a large number of unusual private network policies.

On the other hand, with the source routing of IDPR, the backbone doesn't care what the private network's policies are. The backbone simply advertises its characteristics, and the private network decides whether that backbone is appropriate for a given transmission. Since the private net explicitly states what the path is in the source route setup, the backbones do not need to calculate paths on behalf of the private nets—they don't need to know the private network policies.

Another advantage of IDPR is flexibility in addressing. With source routing, the packets do not contain addresses per se. They contain path descriptions. (Actually, a route setup packet contains the path description, and subsequent packets contain only a path identifier.) The information that describes the location of a private network with respect to the backbones, which is normally contained in the hierarchical address (for instance, with *IDRP*), is now encoded in the topology information carried in the IDPR routing updates. Because this information is not locked into a fixed-size address, it can be used more flexibly.

An interesting characteristic of IDPR source routing is increased flexibility in topology dissemination, resulting in different and potentially beneficial scaling characteristics. Since each private domain calculates its own routes, it only needs to know the topology that it actually uses. Scaling benefits are therefore possible because the private domain need not store topology it doesn't care about. If the private domain needs to send packets to an unexpected destination, and doesn't have the appropriate topology stored locally, it can send a query to a route server that contains more topology information.

continued on next page

Inter-domain Routing in the Internet *(continued)*

The server in turn can calculate a path on behalf of the private domain, or send the private domain the appropriate topology information. Also, since backbones don't calculate routes, they do not need to maintain the global topology, resulting in more scaling benefits.

However, IDPR alone does not completely escape scaling problems, since the route servers that answer queries must themselves maintain complete topology information. Further, different scaling problems are introduced. The reduction in stored information is offset by the route server queries. While one assumes that information will be stored so as to minimize the number of queries, variations in traffic (away from what was expected) may result in large numbers of queries, and the resulting overhead and delay in path setup.

Another interesting aspect of IDPR source routing is its reaction to failures. With *IDRP,* a failure results in execution of the distributed routing algorithm. In some cases, this does not amount to significant activity, because the failure might be localized to one part of the hierarchy. But if, for instance, the failure partitions a high-level backbone, there could be many routing updates. With IDPR, failure notification is limited to routers that have set up or try to set up a path over the failed resource. This may amount to more or less update activity compared to IDRP, depending on the location of the failure and the number of paths set up through the failure.

Conclusions

As usual, there are no hard and fast conclusions. On the one hand, IDPR gives more policy control than *IDRP.* On the other, IDPR 1) is more complex than *IDRP,* and 2) has less predictable overhead. In particular, the route setup and resetup (in case of failure) and route server query functions are significant added complexities.

Making the overhead of IDPR less predictable is the fact that it relies on the caching of route setup information. The more paths setup at a particular time, the more overhead for the router, since the router must store the mappings between path id and next hop for every path. The traffic characteristics for data are volatile compared to those for voice. Efficient operation of IDPR depends on each user communicating to a relatively small number of other users (therefore resulting in a small number of setup paths). This discourages applications where users talk to a large number of other users, such as resource discovery applications. Since it is hard to predict future applications, there is uncertainty about the future efficiency of IDPR.

Ultimately, preference of the two approaches depends on one's requirements. If one believes that basic policies, such as being able to choose between multiple backbone systems, are sufficient then the disadvantages of IDPR outweigh the advantages. If one believes that sophisticated, user-unique policies are required, then IDPR seems advantageous.

It is difficult if not impossible to come to consensus on policy requirements. It is therefore difficult to reach consensus on whether IDPR or *IDRP* is preferable. However, the question of consensus may be moot. The previous version of *IDRP* (BGP2) is already implemented in parts of the Internet. This establishes a basis from which to evolve to *IDRP.* Further, *IDRP* seems to have the inside track in ISO, giving it still more clout. Also going against IDPR is the fact that currently the Internet is based on incremental routing.

Switching to IDPR-style source routing is more difficult than continuing with *IDRP*-style incremental routing. Finally, since *IDRP* scales adequately and satisfies the basic policy requirements, there may be little incentive to transition to IDPR, even if it becomes widely perceived as superior to *IDRP*.

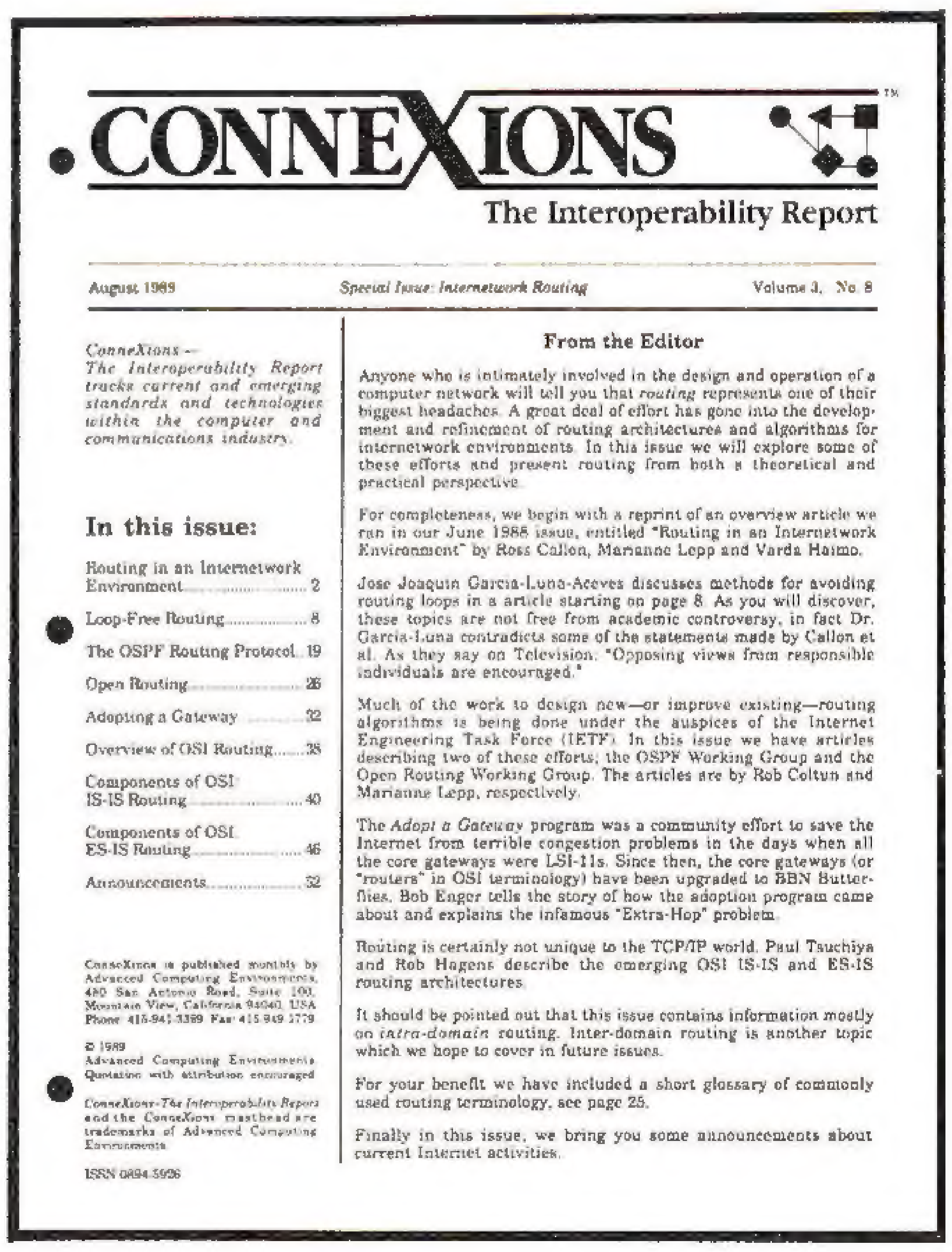
It may be possible that certain communities of private networks and backbones require the policy control of IDPR, while the rest of the world is satisfied with *IDRP*. This raises the spectre of interoperability between IDPR and *IDRP*. On the surface, there doesn't appear to be any reason why the two can't be made to interoperate. In particular, a group of domains running either IDPR or *IDRP* can always be made to appear externally as a single routing domain.

Almost certainly, the politics of IP and ISO standards will determine the fate of IDPR and *IDRP*. However, both provide the functions of scaling and policy needed to advance the people power revolution promised by internet technology.

PAUL TSUCHIYA received his BSEE (1980) from Colorado State University. From 1982 to 1989 he was a Member of the Technical Staff at MITRE in McLean VA. Now, he is a Member of the Technical Staff in the Applied Research Area of Bell Communications Research. In both jobs, Mr. Tsuchiya has concentrated in the area of data network routing, particularly routing for large networks. Mr. Tsuchiya was a member of the Open Routing Working Group, where IDPR is being developed, and is a member of ANSI X3S3.3, where *IDRP* is being developed. He can be reached as `tsuchiya@thumper.bellcore.com`.

The other Routing Issue

Intra-Domain Routing was covered in *ConneXions* Volume 3, No. 8, August 1989, *Special Issue on Internetwork Routing.* This 56-page report contains many articles on TCP/IP and OSI routing protocols written by experts in the field. Order yours today to ensure that you get the complete picture. This and all other back issues of *ConneXions* are available for purchase. Call us at 1-415-941-3399 or toll-free 1-800-INTEROP and ask for our free index pages to pick the issues of interest to you.



CONNEXIONS™

The Interoperability Report

August 1989 | *Special Issue: Internetwork Routing* | Volume 3, No. 8

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Advanced Computing Environments, 480 San Antonio Road, Suite 100, Mountain View, California 94040, USA
Phone 415-941-3399 Fax 415-949-1779

© 1989
Advanced Computing Environments
Quotation with attribution encouraged

ConneXions–The Interoperability Report and the *ConneXions* masthead are trademarks of Advanced Computing Environments

ISSN 0894-5926

From the Editor

Anyone who is intimately involved in the design and operation of a computer network will tell you that *routing* represents one of their biggest headaches. A great deal of effort has gone into the development and refinement of routing architectures and algorithms for internetwork environments. In this issue we will explore some of these efforts and present routing from both a theoretical and practical perspective.

For completeness, we begin with a reprint of an overview article we ran in our June 1988 issue, entitled "Routing in an Internetwork Environment" by Ross Callon, Marianne Lepp and Varda Haimo.

Jose Joaquin Garcia-Luna-Aceves discusses methods for avoiding routing loops in a article starting on page 8. As you will discover, these topics are not free from academic controveray, in fact Dr. Garcia-Luna contradicts some of the statements made by Callon et al. As they say on Television: "Opposing views from responsible individuals are encouraged."

Much of the work to design new—or improve existing—routing algorithms is being done under the auspices of the Internet Engineering Task Force (IETF). In this issue we have articles describing two of these efforts; the OSPF Working Group and the Open Routing Working Group. The articles are by Rob Coltun and Marianne Lepp, respectively.

The *Adopt a Gateway* program was a community effort to save the Internet from terrible congestion problems in the days when all the core gateways were LSI-11s. Since then, the core gateways (or "routers" in OSI terminology) have been upgraded to BBN Butterflies. Bob Enger tells the story of how the adoption program came about and explains the infamous "Extra-Hop" problem.

Routing is certainly not unique to the TCP/IP world. Paul Tsuchiya and Rob Hagens describe the emerging OSI IS-IS and ES-IS routing architectures.

It should be pointed out that this issue contains information mostly on *intra-domain* routing. Inter-domain routing is another topic which we hope to cover in future issues.

For your benefit we have included a short glossary of commonly used routing terminology, see page 25.

Finally in this issue, we bring you some announcements about current Internet activities.

Issues in Inter-domain Routing

by Robert Woodburn, SAIC

Introduction

Sitting at my workstation with my lips tightly pursed and my eyes starting to get blurry from straining, I try for the umpteenth time to *ping* BBN.COM. It was reachable just this morning and I have this *real* urgent message that I need to mail off to the other folks in the working group before tomorrow's video conference. I run *traceroute* (also for the umpteenth time) to try and find the offending link. As the number of hops progresses into the double digits, I think back to just a few years ago when internetworking seemed so much simpler and we and BBN both were connected to the ARPANET backbone. Networking is pretty simple when everybody shares a common network. But now there are multiple backbones, scattered regional networks connected to the backbones and lots and lots of stub networks connecting to regionals, each other, and just about anything else that can speak the Internet Protocol.

You probably have experienced a lot of the same frustrations described above, and you may have noticed that things are getting worse. Well, the reality of the matter is that things are getting worse. Exponentially worse. But the good news is, before you go and throw your workstation out the window, that there are solutions progressing in the wings. The problem is routing within very large, heterogeneous, and administratively diverse networks. The solution may be found in work currently progressing in inter-domain routing.

Growth

The Internet today faces tremendous growth. That really should read GROWTH. In its humble beginnings, the Internet consisted of some tens of networks. Today, there are nearly 1800 networks connected to the Internet, and another order of magnitude of network numbers have been assigned by the *DDN Network Information Center*. The reason that having so many networks in the Internet is a problem, is that calculating routing paths can be as bad as an $O(N^3)$ problem. The growth to date has been exponential, with the number of networks nearly doubling every year. There has even been talk about running out of IP addresses in the next decade, and creative ways of efficiently using the remaining numbers have been proposed [18]. Needless to say, a *critical* situation is arising.

Tradeoffs in routing

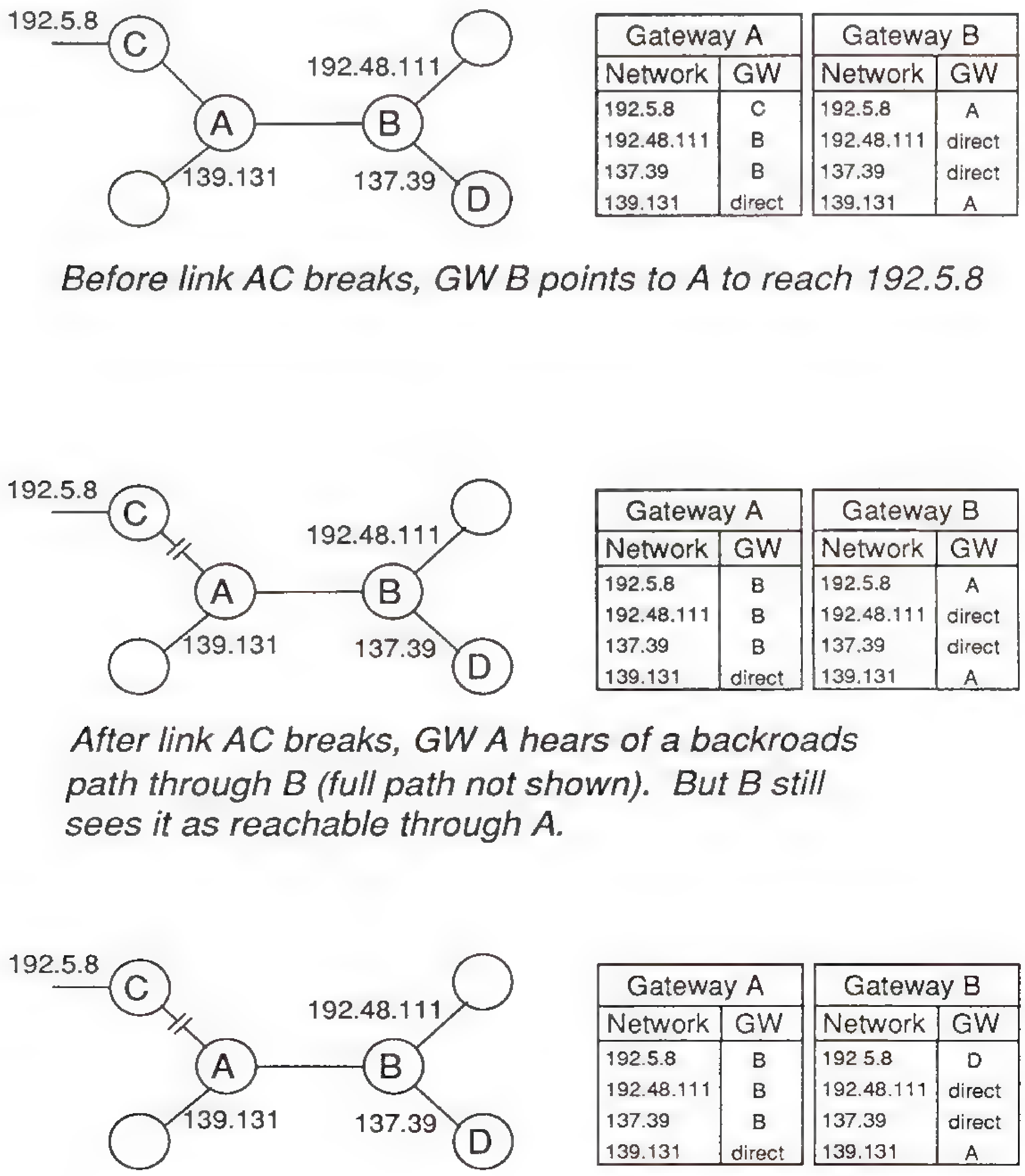
When designing any routing paradigm, there are many tradeoffs that must be considered. The first is between routing complexity and the speed of the routing calculation. Certainly a desirable goal for routing is the generation of optimal, (meaning the best route for the type of traffic offered) efficient routes which take into account changes in topology, the type of service required by the user, the type of service provided by the resources, the current loading of available resources, and the whims of network administrators who control the resources along the path. As all of these "metrics" are factored into the calculation, the processing burden rapidly increases.

There is a tradeoff between using a global database that includes all possible networks in the system and a localized database that applies only to a certain area within the entire system. The advantage of using a global database, is that all possible paths can be calculated in their entirety between any two points. The disadvantage is that as the database grows it can take a very long time to find the best paths. A localalized database helps solve this problem by only worrying about networks within a certain area. Of course, this means that endpoints outside the area would be unreachable, or at least complete paths could not be calculated for them.

Centralized versus distributed algorithms

Another tradeoff is between a *centralized* algorithm and a *distributed* algorithm. There are problems with both. A machine running a centralized algorithm must have the entire routing topology in its database. It must have a means of collecting topological information for all networks and gateways. This is can be accomplished through a "link-state" database. This database describes the condition of the link between two nodes. With such a database a machine could calculate routes between any source and destination pair given any type of service or administrative policy constraints, in addition to calculating routes with itself as the origin. As such it could behave as a *route server* for others needing routes. The problem is that for this to operate over the entire Internet, there would be an incredible amount of traffic flying around. All gateways would have to forward their local link-state information to the route server. Thus for a large system, a centralized algorithm does not scale well.

A distributed algorithm takes care of some of these problems, but introduces others. In this scheme, each gateway involved in the routing algorithm could use a link-state database, or what is called a "distance-vector" database. In the latter scheme, a gateway reports the reachability of its directly connected networks and acquires the reachability of networks connected to its neighbors. In this way it builds a reachability database for all known networks. The gateway would not know all of the points along the path for a given route. It would only know which neighbor to forward a packet to for a given destination.



Gateway A		Gateway B	
Network	GW	Network	GW
192.5.8	C	192.5.8	A
192.48.111	B	192.48.111	direct
137.39	B	137.39	direct
139.131	direct	139.131	A

Before link AC breaks, GW B points to A to reach 192.5.8

Gateway A		Gateway B	
Network	GW	Network	GW
192.5.8	B	192.5.8	A
192.48.111	B	192.48.111	direct
137.39	B	137.39	direct
139.131	direct	139.131	A

After link AC breaks, GW A hears of a backroads path through B (full path not shown). But B still sees it as reachable through A.

Gateway A		Gateway B	
Network	GW	Network	GW
192.5.8	B	192 5.8	D
192.48.111	B	192.48.111	direct
137.39	B	137.39	direct
139.131	direct	139.131	A

B finally receives a routing update and points to the backroads path through D.

Figure 1: Example of how a routing loop can form

continued on next page

Issues in Inter-domain Routing *(continued)*

Note that this reachability database would only be valid for the metric used in defining reachability. The neighbor that is best for high throughput traffic (e.g., over a satellite link) might very well be different than the neighbor that is best for low delay traffic (e.g., over slower, but lower delay terrestrial lines). This would require a separate database for each metric and could become unmanageable if there were more than just a few metrics to consider.

An additional problem with a distributed algorithm is how changes propagate through such a system. In the centralized routing, there is only one database. If a link changed somewhere and the database became outdated, the route would fail and data sent on that path might be "blackholed." In the distributed case, however, if something changes in the database, routing loops could be introduced while the change propagates. This is a result of every gateway having an independent database, but one that only points as far as the next hop (the next gateway address).

As illustrated in Figure 1, it might be possible that as a result of a link changing state, gateway A would believe that gateway B is the best place to send packets for a certain destination. But before the change, B may have believed that A was the best place to send packets for that destination, hence a routing loop. It may take several minutes for gateway B to receive the update for the change in state, and the loop will disappear once it does. Thus, these loops eventually can work themselves out, but in a system where the number of networks is growing exponentially, both the number of changes and the amount of time required for this information to propagate grow proportionately. In other words, a distributed algorithm does not scale well either.

Current Inter-Domain Routing

Now, hold on a second, put the workstation back down and get away from the window. There are still some other alternatives. So far we have assumed that there was a single flat space over which to perform routing calculations. This restriction requires that every gateway know everything in order to get anywhere. In the real world, all gateways are not created equal and the topology is not quite so flat.

Administrative Domains

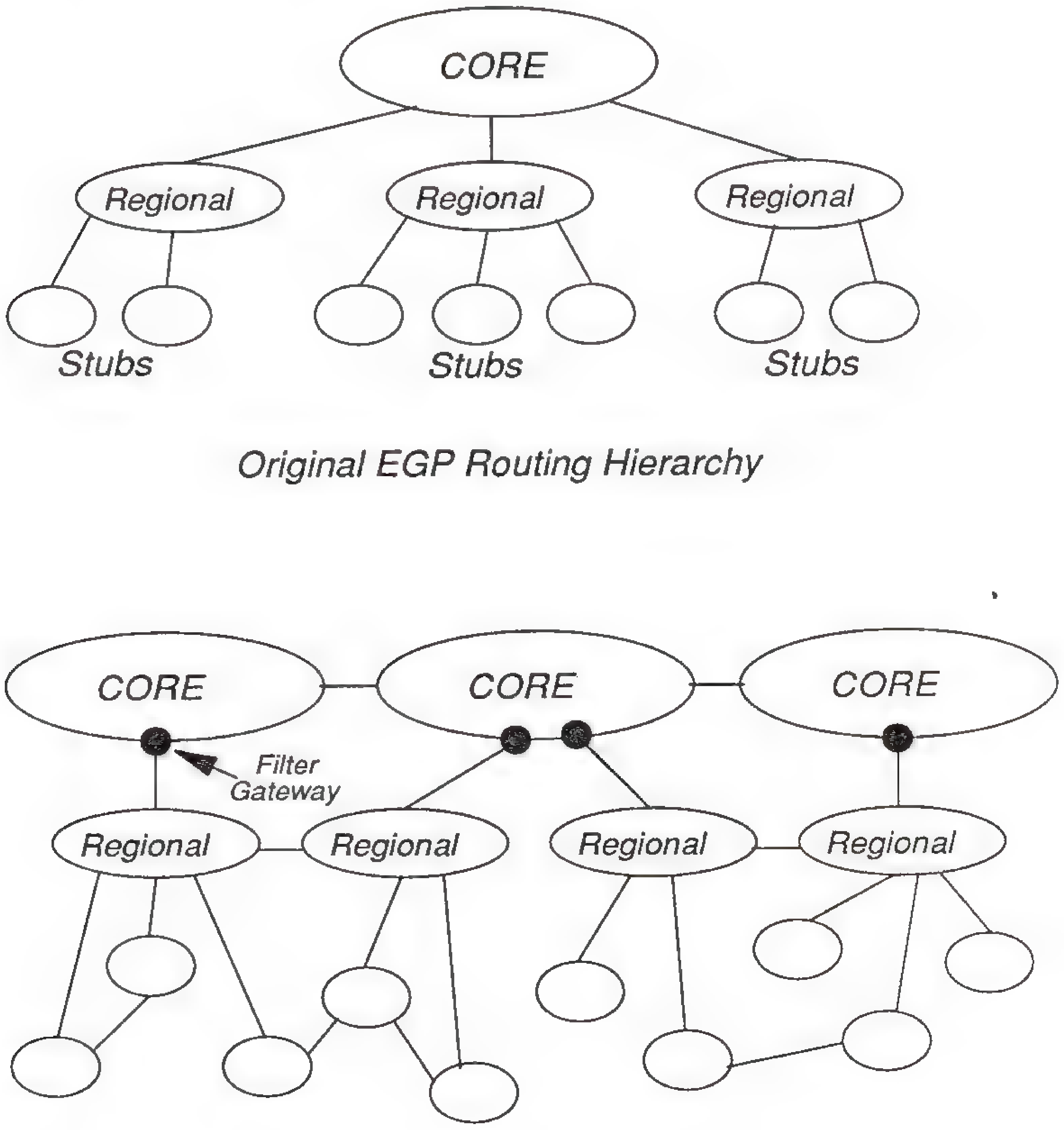
The Internet can be viewed as consisting of *Administrative Domains* (AD). An AD is a region of control administered by a single authority. These regions have been known as "Autonomous Systems" in the Internet community. Networks, gateways, and routing protocols within the AD are all operated by this authority. Gateways within the AD are called *Interior Gateways* and run a common routing protocol called an *Interior Gateway Protocol* (IGP) of which several are currently being used. Gateways outside of the AD are *Exterior Gateways.* In order to communicate with these gateways an *Exterior Gateway Protocol* (EGP) is used in gateways at the edges of the AD.

The core system

When the first EGP [13], was used in the Internet, there was a tree hierarchy implicit in its design. There was a single *core* AD which acted as the root of the tree, several "mid-level" ADs, and finally some leaves on the tree which use the mid-level ADs for getting to the core [2, 3]. Routing in this hierarchy is greatly simplified. The leaves of the tree route to their mid-level AD when they don't know how to find a destination. The mid-level, in turn, would route to the core. The core would have to know everything about all connected networks and would route packets back out to the leaves. This means that routes may not be optimal, but routing becomes a solvable problem.

EGP limitations

This all sounds well and good, but EGP has limitations on how it handles information being passed around. It cannot accept "third-party" information. Thus, the tree could never grow to more than three levels. Another problem is that the core still has to know everything, which means as the Internet continues to grow there have to be super-gateways in the core. To make matters far worse, the Internet is not growing as a tree [17]! It looks more like a misbehaved patch of ivy. The tree has become a mesh with many cross connections between mid-level regional ADs and their leaves, or stubs, as in Figure 2. This has been handled by introducing filtering of the routing information being passed around. The core and mid-level ADs will only "believe" information conforming to their filters. These filters are hand-configured tables in "firewall" gateways, the black dots in Figure 2, that specify which networks can be reported by whom, how they are reported, and what is done with them in the routing database. The result is that if your network is not in the core's filters, then you will not be able to route through the core at all (which was why I could not reach BBN back at the beginning, my network had been dropped out of the core's filter and the return path from BBN failed). The fact that these tables are hand configured can cause a great deal of the dynamic information about a path to be lost. Because the tree architecture no longer applies, there must be a new EGP that works well for a mesh.



Original EGP Routing Hierarchy

Current Internet Reality

Figure 2: Intended Internet hierarchy versus current reality

BGP

There is a great deal of effort going into fixing the growth problem and several approaches are being used. One problem with EGP has been the incredible size of update messages as the number of networks increases. The *Border Gateway Protocol* (BGP) [12] is currently being deployed in several places within the Internet.

continued on next page

Issues in Inter-domain Routing *(continued)*

BGP solves the problem of the large update messages by using incremental updates. It makes sense to only update those parts of the routing database that change with time, rather than sending a complete database on a periodic basis. BGP helps with the problem of routing loops by looking carefully at the information received from several autonomous systems and detecting multiple occurrences of an AD in a path. These loops can be reported to other ADs and removed from the routing database. Additionally, BGP provides the capability of selecting paths preferentially. The protocol can be configured so that paths with particular characteristics are chosen over other paths. Some of these characteristics include: the number of ADs in a path, the presence or absence of a particular AD in a path, stable versus unstable paths [8].

Inter-Domain Routing

The problem of routing within an AD has grown as the ADs have grown, but recently deployed IGPs have begun making things far more manageable. The problem of routing between ADs, however, is becoming totally unmanageable. Enter in *Inter-Domain Routing.* If the ADs were seen as nodes in an inter-domain "network" then the current routing technologies could be applied at this level. With this additional level of routing would come an additional level of addressing. The problem is no longer what networks to traverse to reach a given IP address, but what domains to traverse to reach a destination AD. This essentially adds a level of routing hierarchy which can be a mesh on top of an existing mesh, an entire virtual network. This does create some difficulties, however.

Addressing

The first and most obvious problem is addressing the entities in the system. Machines on the Internet currently speak IP and understand IP addresses, they have no concept of AD addresses. There would have to be some entity who could speak IP and also see things at the inter-domain level. There would also have to be a mechanism for translating destination IP addresses into AD addresses. This is similar to the need for a mechanism for translating from IP addresses to a physical address, like an Ethernet address. Once the mapping was available, the inter-domain entity could begin forwarding the packet on its way using some routing mechanism based on existing technologies. A possible option would be to change the length of IP addresses to include a domain identifier for all entities in a domain. If and when OSI protocols become commonplace, then much longer addresses will be available [4]. But changing the IP addresses would require changing all hosts, not a very appealing option in the near term.

Another problem is getting a datagram "over the wall" between ADs. An entity at the inter-domain level would snag a datagram at the IP level of its domain, translate it to the inter-domain space, and forward it to some next inter-domain hop. In practice, IP would have to be used to reach the next inter-domain hop. Thus, when forwarding the datagram, the inter-domain entity would translate the next hop at the inter-domain level to an IP address and send it off like any other IP datagram. The only requirement is that the next hop be reachable via the local IP routing protocols. Since the IGP of an AD only operates over that AD without necessarily knowing anything about adjacent domains, knowing the next hop's IP address could be a problem. One approach is to place the inter-domain entities at the edges of the domain and have an interface that would be able to talk over some sort of wire to a similar entity in an adjacent domain. These two entities would not share IGP information, only inter-domain information and datagrams to be forwarded.

A problem of policy

Finally, there is a problem with *policies*. As more and more organizations connect to the Internet and the issues of who is going to pay for what are being considered, network administrators have to begin worrying about how they are going to restrict access to their network resources. A policy could be something like, "We'll act as a transit domain for educational traffic, but anything commercial gets dropped on the floor."

There is no rhyme or reason to policy, it depends entirely on who is administering the domain and what the mission of the domain is. Currently no IGP supports a flexible concept of policy. One possible means of handling multiple policies would be to provide a separate address for each policy being maintained [23]. In this scheme an administrator would assign a different address for each means of reaching a destination. A high throughput path would be taken to reach the destination's "high throughput" address. A low delay path would be taken to reach the destinations "low delay" address.

A means of representing policy has been described by Dave Clark [5] and some examples of policies can be found in [1, 6, 9]. How to incorporate policy into routing is a difficult problem. Conceptually, a separate routing table might be required for every different policy, just as for every different type of service. Clearly the amount of information explodes.

IDPR

The *Open Routing Working Group* (ORWG) of the IETF is currently developing an inter-domain routing protocol called the *Inter-Domain Policy Routing Protocol* (IDPR) [7, 10, 11, 16]. The inter-domain entities in the IDPR are called *Policy Gateways* (PG). Two policy gateways which are directly connected to each other, each belonging to a different AD, form a *Virtual Gateway* (VG). Several PGs can be part of a VG, but the restriction is that only two ADs can be involved. When routes are calculated, they are formed in terms of the ADs and the VGs and some policy information.

Source routing

The protocol uses *source routing* (not to be confused with IP source routing!) as the means of forwarding packets. This means that the PG trying to reach another AD, must obtain a route specifying each inter-domain hop along the entire path at the AD level over which the datagram is to be forwarded. Note that the hops internal to the AD are not calculated, since they are not seen at the inter-domain level.

Once the path is known, subsequent datagrams are sent along the same path. By requiring the source to calculate the route, it is possible to be very flexible in the types of policies and services that can be specified in the routing database.

The drawback of this approach is that this route might change with time and the source will have to calculate a new path when the original one fails. If the number of domains remains small, however, and the number of PGs are kept small as well, then these path changes should be relatively infrequent. The immediate question this should raise is, "what about the growth problem, won't the number of domains eventually explode as well?" This is true, but the intention is to develop an architecture that is extensible. The IDPR has a concept of groupings of ADs called "super-ADs" that will help reduce this problem. In addition, there is the possibility of overlaying the architecture multiple times when things become unmanageable.

continued on next page

Issues in Inter-domain Routing *(continued)*

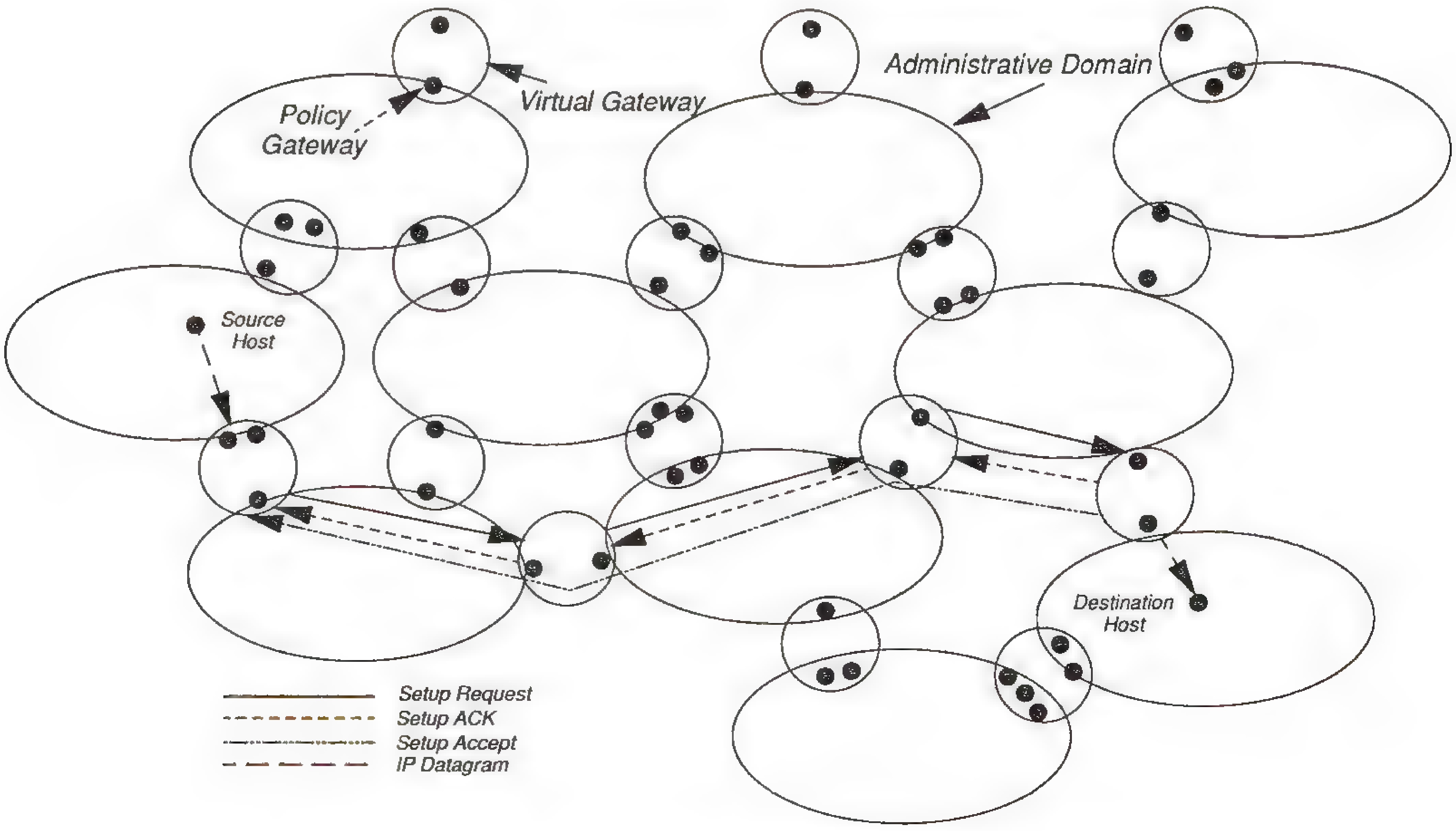
Encapsulation

The mechanism used to get datagrams into the inter-domain level is *encapsulation.* An IP datagram that is received by a PG is examined to see if the destination address is in the local domain. If it is, the datagram is routed normally. If not it is passed up to the inter-domain level. At the inter-domain level, a path is determined for the datagram by an entity called a *Route Server* (RS), and then the path is "setup" in all of the PGs along the path to the destination. The path setup operation is accomplished through a separate path setup protocol that informs the PGs about the new path and verifies the correctness of the path in terms of connectivity and policy. The flow of information is illustrated in Figure 3.

Once the path is set up, the source PG encapsulates the original IP datagram in a new IP header along with some inter-domain routing information, namely a flow identifier naming the path to be used, and sends it to a PG in an adjacent domain. When the packet is received by the PG in the next domain, the flow identifier is matched to the path established by the path setup protocol, and the datagram is sent to the next PG along the path using the local IGP to route the encapsulated message as a normal IP datagram, thus transiting the domain. When the packet finally reaches the destination AD through the last PG on the path, the packet is decapsulated and dumped into the local IP to reach its final destination.

Problems

There are some problems associated with encapsulation. The most serious is the effect it has on error messages that might be generated at some intermediate point when a packet is transiting a domain. The last PG will be seen as the source of the offending datagram and will get sent an ICMP message. This ICMP message will have part of the original inter-domain header included, but it is not clear that there will be enough information for this ICMP message to reach back through the inter-domain gateways to the actual source host, in a form useful to the source. This problem is still under investigation.



Packets from the source host are encapsulated at VGs and sent along the inter-domain path, being decapsulated and reencapsulated at each VG on the path.

Figure 3: Example of the operation of encapsulation seeded by the IDPR setup protocol

Another problem would be the *Time To Live* (TTL) field in the original datagram. Normally this field gets decremented at each hop as it passes from gateway to gateway. With encapsulated packets, this field would not get modified while the packet transits a domain. It might be possible that intervening PGs along the way could tweak this field, but gateways inside the domains do not understand the inter-domain protocol, and would be oblivious to the encapsulated TTL.

Advantages

A great advantage of encapsulation is that conceptually one can keep encapsulating through any number of levels of hierarchy. Another advantage is that the IDPR setup protocol is only one means of "seeding" intermediate hops with the path information. If one wished to use OSPF [13] or some other IGP to pass the connectivity information of the domains, it would be possible. Instead of network numbers, ADs would be used. A further advantage to encapsulation, is that it will be possible to transit domains which may not speak IP. If a domain is using OSI protocols, it will be possible to encapsulate a datagram in an OSI header and allow it to transit the domain as if it were just another OSI datagram.

This is an attractive solution to inter-domain routing. It provides a means of routing between any points in the Internet without IP gateways having to know the entire topology at the IP level. Additionally, it is extensible. If the number of domains does increase to a point where they are unmanageable, then an inter-inter-domain hierarchy could be established on top of the inter-domain level. All government domains might become one internet with its own inter-domain routing structure. To reach the education internet, it might have to go up to the inter-inter-domain level. Some domain at the "edge" of the government internet would have a gateway that could see up to the inter-inter-domain level and would handle passing the datagram to the next adjacent inter-domain to reach the final destination.

Conclusion

So things continue to change in the Internet. More and more networks mean new and different ways of getting around. But don't worry, whenever things start to break in the Internet, there is usually enough noise generated to fix things before they get too bad. Hopefully the work that is progressing in inter-domain routing will keep pace with what lies ahead.

References

[1] Braun, H.W., "Models of Policy Based Routing," RFC 1104.

[2] Braun, H.W., "NSFNET Routing Architecture," RFC 1095.

[3] Brim, S., "IP Routing Between U.S. Government Agency Backbones and Other Networks," Cornell University.

[4] Callon, R. & Braun, H. W., "Guidelines for the use of Internet-IP Addresses in the ISO Connectionless-Mode Network Protocol," RFC 1069.

[5] Clark, D., "Policy Routing in Internet Protocols," RFC 1102.

[6] ECMA, "Inter-Domain Intermediate Systems Routing," Draft Technical Report ECMA TR/ISR, 7th Draft, January 1989.

[7] Estrin, D., "Requirements for Policy Based routing in the Research Internet," RFC 1125.

continued on next page

Issues in Inter-domain Routing *(continued)*

[8] Interconnectivity Working Group, "Application of the Border Gateway Protocol in the Internet," RFC 1164.

[9] Leiner, B., "Policy Issues in Interconnecting Networks," RFC 1124.

[10] Lepp, M., Steenstrup, M., "An Architecture for Inter-domain Policy Routing," Internet Draft, BBN, February 1990.

[11] Little, M., "Goals and Functional Requirements for Inter-Autonomous System Routing," RFC 1126.

[12] Lougheed, K.; Rekhter, Y., "A Border Gateway Protocol (BGP)," RFC 1163.

[13] Mills, D. L., "Exterior Gateway Protocol Formal Specification," RFC 904.

[14] Mills, D. L., "Autonomous Confederations," RFC 975.

[15] Moy, J., "The OSPF Specification, Version 2," Internet Draft, June 1990.

[16] Open Routing Working Group. "Inter-domain Policy Routing Protocol Specification and Usage: Version 1," Draft, April 1990.

[17] Rekhter, Y., "EGP and policy based routing in the new NSFNET backbone," February 1989.

[18] Tsuchiya, P., "Efficient and Flexible Hierarchical Address Assignment," Draft, Bell Communications Research, July 1990.

[19] *ConneXions,* Volume 3, No. 8, August 1989: *Special Issue on Internetwork Routing.*

[20] Dern, D., "Standards for Interior Gateway Routing Protocols," *ConneXions,* Volume 4, No. 7, July 1990.

[21] Tsuchiya, P., "Inter-domain Routing in the Internet—The People Power Revolution Continues," *ConneXions,* Volume 5, No. 1, January 1991.

[22] Y. Rekhter & D. Katz, "The Border Gateway Protocol (BGP)," *ConneXions,* Volume 5, No. 1, January 1991.

[23] P. Tsuchiya, "Scaling and Policy Using Multiple Hierarchical Addresses," Bellcore, 1990.

ROBERT WOODBURN received his B.S. (1986) in Electrical Engineering from Carnegie-Mellon University. Since then he has worked for a systems engineering firm (formerly M/A-COM Linkabit), which has become a division of Science Applications International Corporation (SAIC), on a variety of hardware and software projects, including implementation of the Dissimilar Gateway Protocol and research in OSI network management. He currently is working in the Open Routing Working Group of the IETF to develop an inter-domain routing protocol.

An Introduction to Policy Routing

by Martha Steenstrup, BBN Communications

Administrative Domains

As data communications technologies evolve and user populations grow, the demand for internetworking increases. Internetworks usually proliferate through interconnection of autonomous, heterogeneous networks administered by separate authorities. A collection of hosts, gateways, and contiguous networks governed by a single administrative authority is called an *administrative domain* (AD).

Interconnection of administrative domains can broaden the range of network services available in an internetwork. However, it can also prompt administrative authorities to establish more stringent network service restrictions and user service requirements. AD authorities impose such constraints in order to maintain control over the use of their network resources and the services provided to their users' traffic.

An internetwork composed of diverse administrative domains requires *policy routing* to transport traffic between source and destination domains. Policy routing constitutes route generation and message forwarding procedures that accommodate source and transit policy constraints. The resulting policy routes simultaneously satisfy source user service requirements and obey access restrictions stipulated by the intermediate domains. In this article, we provide a basic introduction to the spectrum of approaches to policy routing.

Policy

Each administrative authority sets *transit policies* that specify how and by whom the resources in its domain should be used. Transit policies define offered service and comprise:

- *Quality:* e.g., delay, throughput, and error characteristics;
- *Access restrictions:* e.g., applied to traffic between certain ADs or classes of users; and
- *Cost:* e.g., charge per byte, message, or session time.

Each administrative authority also influences *source policies* that apply when selecting a route to a given destination. Source policies specify desired quality and cost of service as well as transit domains to favor or avoid.

Domain Interconnectivity

AD policy constraints affect domain interconnectivity. The degree of transit service offered by each administrative domain suggests a natural AD ordering: a hierarchical internetwork topology of backbone networks, regional and metropolitan networks, and campus networks. However, this ordering is not necessarily a partial ordering, as it may not be transitive. For example:

- AD X, AD Y, and AD Z are connected in series;
- AD Y is willing to act as a transit domain for AD Z;
- AD X is willing to act as a transit domain for AD Y but not for AD Z.

Moreover, technological, economical, and political incentives often prompt the introduction of bypass links, augmenting the basic hierarchy. For example, administrative domains at a given level or at different levels in the hierarchy may mutually agree to support special connections between them. These connections provide communications paths outside of those within the hierarchy and are available only to a limited number of administrative domains.

continued on next page

Introduction to Policy Routing *(continued)*

Approaches to Policy Routing

The various approaches to policy routing can be distinguished according to the routing information: namely its contents, the extent of its distribution, and the manner in which it is used for route generation and message forwarding.

The routing information generated and distributed by an administrative domain may consist of either:

- *Distance vectors:* destination reachability according to policy, from the given domain's perspective; or
- *Link states:* the set of transit policies that apply to the given domain and the connectivity to adjacent domains.

Route Generation

In general, distance-vector routing procedures require distribution of routing information to direct neighbors only, while link-state routing procedures entail wide distribution of routing information.

The distance-vector approach is attractive, because it is a distributed route generation procedure. Hence, it consumes only a moderate amount of overhead in computing routes and in distributing and storing routing information. However, the distance-vector approach has the drawbacks of routing loop formation and slow convergence rate, unless administrative domains limit distribution of routing information or distribute complete routes and not only reachability information.

From the policy routing perspective, a more serious problem with the distance-vector approach is how to capture source policies in the routes. Although a domain may select a route according to its source policies from a set of routes distributed by its neighbors, it plays no part in construction of these routes. Intermediate domains construct the routes hop by hop and may select routes which are inconsistent with a domain's source policies. In the worst case, a source may be left with no compatible routes to a given destination, even though such a route exists.

The distance-vector approach can guarantee construction of compatible routes for a source, only by maintaining source policies for all domains at each administrative domain and by selecting routes and forwarding routing information according to which source may wish to use it. This modification increases the storage, distribution, and computation overhead of the distance-vector approach and also implies that source policies must be advertised.

The link-state approach is well-suited to policy routing. It gives sources complete control of route generation using the transit policies distributed by each administrative domain, and it allows domains to keep their source policies private. Accounting for source policies, the link-state approach and the distance-vector approach consume approximately equal amounts of overhead, in terms of computation of routes and distribution and storage of routing information.

Routing Information Distribution

Administrative domains can exercise further control over the use of their resources by restricting the use of their routing information in route generation. There are two ways to accomplish this: restrict distribution of routing information or distribute access restrictions as part of routing information.

With the distance-vector approach, one controls the flow of routing information in the internetwork by distributing routing information according to a set of directed graphs superimposed on the internetwork topology. For each destination domain, there exists an associated directed graph that determines the flow of routing information relating to that destination.

A side effect of using directed graphs is loop-free routing. However, a disadvantage is the potential inability to discover alternate paths when a connection between administrative domains fails. In order to increase the probability of finding alternate routes, the graphs should not be truly directed; some of the links should permit bidirectional flow of routing information.

The main drawback to the directed-graph method for routing information restriction is the effort involved in graph generation and installation in the internetwork, initially and whenever domain interconnectivity or policies change. Graph generation requires a central internetwork coordinating authority and the cooperation of all administrative domains.

With the link-state approach, an administrative domain advertises access restrictions as part of its routing information, in order to limit use of its transit services. Only for reasons of efficiency or privacy will an administrative domain actually restrict distribution of its routing information. Restricting distribution of routing information implies that routing information is no longer consistent across the internetwork. In this case, a source must retain control over message forwarding to a destination, in order to prevent routing loops.

Message Forwarding

There are two alternative methods for message forwarding, once a source selects a route to a destination: *destination-based forwarding* and *source-specified forwarding.* With the destination-based approach, each router makes an independent routing decision based on the destination and on the source service requirements carried in the message. The appeal of this approach is its simplicity and low overhead. However, the disadvantage of destination-based forwarding is that it requires consistent routing information at each router in order to follow the path selected by the source and to avoid the formation of routing loops.

Source-specified forwarding does not require routing information consistency across the internetwork. The source specifies the entire route and intermediate routers forward the message accordingly. This approach gives the source tight control over the path its traffic takes to the destination. However, source-specified forwarding does consume more overhead than destination-based forwarding. Depending on the implementation, the overhead of source-specified forwarding appears as link bandwidth consumption or as processing and memory consumption. In one implementation, each message carries the entire route, while in another implementation a source performs a path setup procedure prior to message forwarding.

With either destination-based or source-specified forwarding, intermediate domains may still reject traffic that is incompatible with their transit policies, in order to protect their resources. Generating routes that account for transit policies reduces the potential of rejecting messages because of policy incompatibility.

continued on next page

Introduction to Policy Routing *(continued)*

Choosing an approach

Before selecting a policy routing procedure for an internetwork, one must answer the following questions about the given approach:

- What type of policy control does it provide?
- What is the overhead in terms of memory, processing, and transmission bandwidth?
- How well does it scale with an increase in the number of administrative domains and policies?
- Does it provide integrity and authentication checks for routing control information?
- Will it operate with different protocol stacks?
- Must hosts participate?
- How does it affect internetwork management?

References

For those interested in obtaining more information about policy routing, we have included the following references of protocols, analyses, and surveys.

[1] ANSI. "Intermediate System to Intermediate System Interdomain Routeing Information Exchange Protocol," Revision 1 of X3S3.3-90-132, June 1990.

[2] H-W. Braun, "Models of Policy Routing," RFC 1004.

[3] L. Breslau and D. Estrin, "Design of Inter-Administrative Domain Routing Protocols," Proceedings of the ACM SIGCOMM '90 Symposium, September 1990.

[4] D. Clark, "Policy Routing in Internet Protocols," RFC 1102.

[5] D. Estrin, "Policy Requirements for Inter Administrative Domain Routing," RFC 1125.

[6] D. Estrin and K. Obraczka. "Connectivity Database Overhead for Inter-Domain Policy Routing," USC Technical Report, TR 90-17, August 1990.

[7] D. Estrin and G. Tsudik. "Security Issues in Policy Routing," Proceedings of the 1989 IEEE Symposium on Security and Privacy, May 1989.

[8] ECMA. "Inter-Domain Intermediate Systems Routing," Technical Report ECMA/TC32-TG10/89/56, May 1989.

[9] J. Honig, D. Katz, M. Mathis, Y. Rekhter, and J. Yu, "Application of the Border Gateway Protocol In the Internet," RFC 1164.

[10] B. Leiner, "Policy Issues in Interconnecting Networks," RFC 1124.

[11] M. Lepp and M. Steenstrup, "An Architecture for Inter-Domain Policy Routing," Internet Draft, February 1990.

[12] M. Little, "Goals and Functional Requirements for Inter-Autonomous System Routing," RFC 1126.

[13] K. Lougheed and Y. Rekhter, "A Border Gateway Protocol (BGP)," RFC 1163.

[14] R. Perlman, "Incorporation of Service Classes into a Network Architecture," Proceedings of the Seventh Data Communications Symposium, October 1981.

[15] SAIC and Ford Aerospace, "Dissimilar Gateway Protocol Specification," CSD-TR1737, September 1989.

[16] M. Steenstrup (Ed.). "Inter-Domain Policy Routing Protocol Specification and Usage: Version 1," Internet Draft, August 1990.

[17] P. Tsuchiya, "Scaling and Policy Using Multiple Hierarchical Addresses," Bellcore, 1990.

[18] *ConneXions,* Volume 3, No. 8, August 1989: *Special Issue on Internetwork Routing.*

[19] D. Dern, "Interior Routing Protocols," *ConneXions,* Volume 4, No. 7, July 1990.

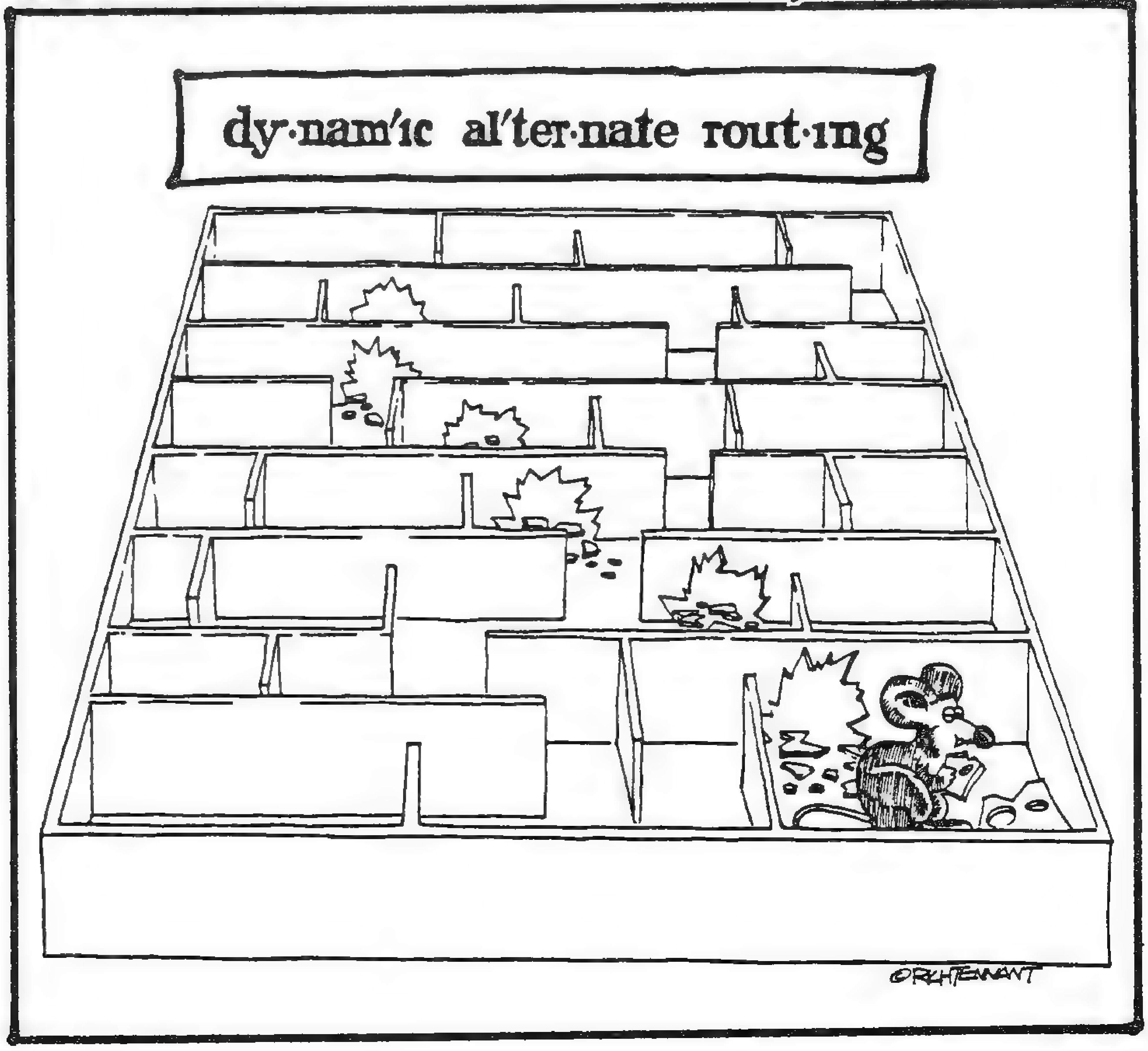
[20] R. Woodburn, "Issue in Inter-domain Routing," *ConneXions,* Volume 5, No. 1, January 1991.

[21] P. Tsuchiya, "Inter-domain Routing in the Internet—The People Power Revolution Continues," *ConneXions,* Volume 5, No. 1, January 1991.

[22] Y. Rekhter & D. Katz, "The Border Gateway Protocol (BGP)," *ConneXions,* Volume 5, No. 1, January 1991.

MARTHA STEENSTRUP works in the Network Analysis Department of BBN Communications in Cambridge, MA. Her interests include design and analysis of routing, flow control, and more generally, distributed adaptive resource management algorithms for communications networks.

The 5th Wave **By Rich Tennant**



The Border Gateway Protocol:

A pragmatic approach to inter-autonomous system routing

by
Yakov Rekhter, T.J. Watson Research Center, IBM Corp.
and
Dave Katz, Merit/NSFNET

Introduction

Inter-autonomous system routing traces its history to the early days of the Internet. To provide connectivity between the ARPANET and the rest of the Internet, BBN introduced the concept of an *autonomous system* (AS). Within an autonomous system, routers use an *interior gateway protocol* (IGP). To exchange routing information between autonomous systems routers that connect multiple autonomous systems together use an *exterior gateway protocol* (EGP).

EGP

The exterior gateway protocol that was implemented in the ARPANET is known as EGP2 [7]. While providing some degree of isolation between routing within an autonomous system and routing between autonomous systems, EGP2 imposes rather unjustifiable restrictions on the interconnection of individual autonomous systems. These restrictions are a result of the nature of EGP2, which was designed as a protocol to exchange reachability information between autonomous systems, rather than as an inter-autonomous system routing protocol. Consequently, the protocol provides no protection against the looping of routing information. The looping problem was addressed in the ARPANET by the introduction of the so-called *core* model. The core model assumed that autonomous systems in the Internet formed a spanning tree with the ARPANET as the root of that tree (the core), thus eliminating any possibility of looping routing information. In other words, means outside of EGP2 (i.e., the engineering of Internet topology) were used to suppress this looping.

Engineered spanning tree

The creation of the NSFNET Backbone in the mid 1980s effectively destroyed the original "core" model. This created ample opportunity for the creation of routing information loops. To address this problem, the second phase of the NSFNET Backbone introduced a new routing architecture [1, 8, 9]. Routing within the Backbone was based on an SPF-based link-state protocol. Routing between the Backbone and the Regional networks was done using EGP2. The major difference between the usage of EGP2 in the ARPANET and in the NSFNET was the concept of an *engineered spanning tree* introduced by the NSFNET. By using configuration databases and filtering EGP information between the Backbone and Regional Networks in both directions, it became possible to enforce (engineer) the spanning tree topology required by EGP2, regardless of the underlying physical topology (which was a mesh). By carefully controlling the exchange of routing information between the Backbone and the Regionals, the new architecture eliminated practically all looping of routing information (although not eliminating transient looping of data packets), and as a consequence achieved highly stable routing.

On the negative side, the engineered spanning tree approach turned out to be overly restrictive compared to a scheme that could dynamically construct spanning trees. Due to its conservative nature and strong emphasis on the suppression of routing information loops, it did not allow the use of some routes that it viewed as potentially prone to looping.

Also, as the number of IP networks known to the NSFNET nearly quadrupled since the inception of the Phase 2 NSFNET, the size of the EGP *Neighbor Reachability* (NR) messages became quite large.

The amount of time it took to process them became non-negligible. Further complications arose from IP fragmentation of the NR messages, since the loss of any fragment would cause the loss of the entire NR message.

To put all this in proper perspective, it is important to understand the reasons why EGP2 was selected as an inter-AS protocol for the second phase of the NSFNET. Experience with the first phase of the NSFNET showed the necessity of separating routing within the Backbone from routing within individual regional networks. Consequently, there was a need for some kind of an exterior gateway protocol. Given the time frame for development and deployment of the second phase of the NSFNET Backbone, the only viable alternative was to use an existing exterior gateway protocol. At that time the only available exterior gateway protocol was EGP2. Thus, the usage of EGP2 in the NSFNET Backbone was largely not a matter of choice, but rather of bare necessity.

From EGP2 to BGP

At the same time as NSFNET adopted EGP2 as its inter-AS protocol, it became increasingly clear that there was an urgent need for much better inter-autonomous system routing than could be provided by EGP2. To satisfy this need, in the summer of 1988, IBM and Merit launched a joint research project on inter-autonomous system routing.

At approximately the same time the Internet Engineering Task Force (IETF) formed the *Interconnectivity Working Group* (IWG), chaired by Guy Almes (Rice University). The mission of this group was to address connectivity issues between the NSFNET Backbone and the Regional networks, as well as between Regional networks. Guy Almes invited several of the key people from IBM and Merit that were involved in inter-AS routing to join the IWG. This was viewed as a very positive move, since the members of the IWG represented almost all of the regional networks, and work on inter-autonomous system routing was as important to the NSFNET Backbone as it was to the regional networks. In addition, the IWG environment provided a broader perspective on the set of problems and requirements for inter-autonomous system routing

Evolutionary approach

Since there was strong pressure to ccme up with a solution for the problems associated with EGP2 as quickly as possible, one of the basic requirements for such a solution was simplicity with respect to implementation. To meet this requirement, initial discussion within the IWG was focused on possible modifications to the existing protocols. Since the only widely implemented protocol was EGP2, members of the IWG spent a fair amount of time looking at possible modifications to EGP2. While some of the proposed modifications would have solved the routing information looping problem, none of them addressed the issue of the large NR messages exchanged by EGP peers.

To solve the problem of large NR messages, it was suggested that EGP2 be replaced with EGP3. However, EGP3 still did not solve the problem of routing information looping, and in addition there were no known implementations of this protocol. Due to the incremental nature of EGP3 updates, implementation of the protocol was viewed as a rather complicated task, thus violating the requirement that it should not be hard to implement.

Revolutionary approach

By the time of the January, 1989 IETF meeting in Austin, it became increasingly clear that any attempt to solve the problems of EGP2 by modifying the existing protocols would be doomed to failure, and that the only possible solution would be to design a new protocol.

continued on next page

The Border Gateway Protocol *(continued)*

Mounting problems associated with EGP2 in the NSFNET and its client networks brought increasing pressure to come up with an immediate solution for inter-autonomous system routing for the NSFNET. This solution had to meet the following four basic requirements:

- It should provide loop-free routing information exchange.
- It should drastically reduce the amount of information that is exchanged between a pair of routers.
- It should be simple enough to implement.
- It should be flexible enough to allow graceful phasing into the existing Internet.

The time that had been spent in the IWG discussing possible modifications to the existing protocols was not wasted. To the contrary, these discussions brought to fruition a set of three simple but powerful ideas. First, to solve the problem of routing information looping, which was quite serious with EGP2 and EGP3, it was proposed that the new protocol carry, in addition to reachability information, the list of autonomous systems that this reachability information had traversed thusfar. Second, to solve the problem of large NR messages, it was proposed that the new protocol use incremental updates, where a pair of routers exchange only changes in the routing information (rather than the whole set of routing information). Third, it was realized that the bulk of the complexity in EGP3 resulted from the built-in reliable transport mechanism that was essential for supporting incremental updates. To preserve incremental updates, but eliminate this complexity, it was suggested that the new protocol run on top of an existing reliable transport protocol—TCP.

BGP early history

During the Austin IETF, over lunch, Len Bosack (cisco Systems), Kirk Lougheed (cisco Systems), and Yakov Rekhter put these three ideas together. They sketched a protocol which they called "A Border Gateway Protocol (BGP)." For the lack of any paper the original version of the protocol was written on three napkins, which gave BGP its unofficial title as a TNP ("Three Napkin Protocol"). Returning from the IETF meeting, Kirk and Yakov expanded the original three napkins into few pages of text, clarified some ambiguities (caused by ketchup stains on the napkins), and in less than a month came up with the first two independent working implementations of the protocol. With significant help from Jessica Yu (Merit/NSFNET), they resolved some interoperability issues, and by the end of spring, 1989, BGP was deployed in the NSFNET Test network.

From the beginning, work on BGP was widely discussed within the IWG. Member of the IWG provided valuable suggestions and comments. In June of 1989, Lougheed and Rekhter published RFC 1105 [5], containing a complete specification of the protocol. In the summer of 1989, Jeff Honig (Cornell University) implemented BGP as part of *gated*. This was the third independent implementation and the first one for which the source code was placed in the public domain.

Defining BGP Architecture

In June of 1989, Hans-Werner Braun (Merit/NSFNET) published RFC 1104 [2], in which he gave a critical review of two possible approaches to inter-AS routing. His paper established the foundation of the basic techniques for policy-based routing that could be supported by BGP.

While RFC 1105 defines BGP as a protocol, the document was silent about how the BGP architecture and protocol could be applied to the Internet. In the summer of 1989, work began on the document that would describe the application of BGP in the Internet. This work was carried by a group of people that included Jeff Honig, Dave Katz, Matt Mathis (PSC), Yakov Rekhter, and Jessica Yu, as well as other members of the IWG. By early 1990 a draft version of this document was completed and published as an Internet Draft entitled "Application of the Border Gateway Protocol in the Internet."

From experimental protocol to Proposed Internet Standard

In parallel with work on the BGP architecture document, Kirk Lougheed and Yakov Rekhter began work on a revision of the original BGP specification. The revised version (known as BGP2) appeared as an Internet Draft simultaneously with the architecture paper. It reflected operational experience with the original BGP protocol, generalized some of the concepts fundamental to BGP, and clarified several issues that were underspecified in the original RFC.

In the spring of 1990, both documents were reviewed by a committee appointed by the Routing Area Director of the IESG, Bob Hinden. As a result of this review, he recommended that BGP advance to the status of Proposed Internet Standard. This recommendation was accepted by the IAB, and in June, 1990, the protocol and architecture documents were published as RFCs with the status of Proposed Internet Standard (RFC 1163 [6], RFC 1164 [3]).

Three independent implementations of BGP2 by cisco, IBM, and by Dennis Ferguson (CA*net) were developed and successfully tested for interoperability within a month after the publication of RFC1163 and RFC 1164.

Current status

As of today, BGP is deployed and operational in the NSFNET and in CA*net. With the creation of the T3 NSFNET Backbone, BGP will be extended to cover it as well as the current backbone. It is planned to extend BGP into EASInet in the near future as well.

Current work with BGP2 focuses on several major areas. In the area of network management, Steve Willis and John Burruss from Wellfleet have developed an experimental MIB for BGP2 [10]. The next step will be to implement this MIB and get some operational experience with it. In the area of authentication we hope to use the approach that is currently being proposed for use in SNMP. Finally, there is the vast unexplored area of possible routing policies that can be supported with BGP. The IWG is focusing on identifying the set of most widely used policies. The outcome of this work will be the set of recommended policies that all BGP routers have to support. Within the next six months we hope that BGP will progress from Proposed Internet Standard to Draft Internet Standard.

As mentioned previously, BGP is the first true inter-AS routing protocol designed for TCP/IP Internets. On the surface, BGP looks like a traditional distance vector protocol, but closer examination reveals rather significant differences.

AS-PATH

To suppress the looping of routing information, BGP introduces the concept of an autonomous system path (AS-PATH). As reachability information traverses the Internet, this information is augmented by the list of autonomous systems that have been traversed thusfar, forming the AS-PATH. The AS-PATH allows straightforward suppression of the looping of routing information. In addition, the AS-PATH serves as a powerful and versatile mechanism for policy-based routing.

continued on next page

The Border Gateway Protocol *(continued)*

Route Attributes

An AS-PATH is just one of what is known in BGP as *Route Attributes* that may be associated with a route. BGP organizes route attributes into the following types: well-known (those that must be recognized by all autonomous systems), and optional (those that some autonomous systems may not recognize). Furthermore, well-known attributes are divided into mandatory attributes (those that must be present with every route), and discretionary attributes (those that do not have to be present with every route). An AS-PATH is an example of well-known mandatory attribute. The notion of route attributes serves as a very powerful mechanism for expandability of the protocol. The provision for optional attributes allows experimentation that may involve a group of BGP routes without affecting the rest of the Internet. New optional attributes can be added to the protocol in much the same fashion as new options are added to the Telnet protocol, for instance.

The BGP version negotiation mechanism provides support for graceful backward compatibility. If a new version of the protocol is developed, machines running BGP will negotiate the highest numbered version of the protocol that both machines support.

The concept of a metric has an extremely limited scope in BGP, since the protocol does not rely on metrics for routing information loop suppression (another departure from distance vector protocols). The only type of metric supported by the protocol is the INTER-AS METRIC attribute, which is used to discriminate multiple links between the same two neighboring autonomous systems. Note that the scope of this metric is local to the originator of the metric (the AS receiving the attribute only makes an unsigned comparison with another metric received from the same AS) and the metric is never combined numerically or propagated beyond the receiving AS.

Incremental updates

To conserve bandwidth and processing power, BGP uses incremental updates, where after an initial exchange of complete routing information, a pair of BGP routers exchange only changes (deltas) to that information. Operational experience with BGP shows a tremendous reduction in the amount of routing traffic and time that is necessary to exchange and process routing information as compared to EGP2. To ensure reliable delivery of routing information (necessary for incremental updates) BGP uses TCP as a transport protocol.

It is important to note that BGP is a self-contained protocol. That is, it specifies how routing information is exchanged both between routers in different autonomous systems, and in routers within a single autonomous system. This differs from EGP2, which specifies only the exchange of routing information between routers in different autonomous systems. To allow graceful coexistence with EGP2, BGP provides support for carrying EGP-derived routes.

Extending the BGP Architecture to OSI

Positive experience gained with BGP suggested the possibility of extending the BGP architecture to the OSI protocol suite. This extension was carried out by Dave Katz, Kirk Lougheed, and Yakov Rekhter who essentially recasted the original BGP protocol specification and architecture document in OSI terminology, and presented it to the ANSI X3S3.3 committee (OSI Network and Transport Layers) in January, 1990, under the name of the *Border Router Protocol* (BRP). BRP was favorably received by the members of the X3S3.3 committee. Using the BRP document as a foundation, X3S3.3 produced the *Inter-Domain Routing Protocol* (IDRP) [4], subsequently approved by ANSI as the US proposal for OSI Inter-Domain Routing, and submitted to ISO for member body comment in September, 1990.

Currently there are several independent efforts underway to implement a subset of IDRP. While both IDRP and BGP are based on a common architecture, there are rather significant differences between them. Among major functional differences are support for hierarchical inter-domain routing and a larger set of path attributes in IDRP.

Conclusion

The Border Gateway Protocol was designed to meet the needs of the operational Internet. As a consequence, it was designed to operate within the current Internet environment. At the same time, the designers of the protocol made its architecture easily extensible. The creation of IDRP serves as a factual proof of this extensibility.

References

[1] Braun, H-W., "The NSFNET Routing Architecture," RFC 1093, February, 1989.

[2] Braun, H-W., "Models of Policy Based Routing," RFC 1104, June, 1989.

[3] Honig, J., D. Katz, M. Mathis, Y. Rekhter, and J. Yu, "Application of the Border Gateway Protocol in the Internet," RFC 1164, June, 1990.

[4] ISO/IEC JTC1/SC6/N6271, Information Processing Systems—Data Communications and Information Exchange Between Systems—Intermediate System to Intermediate System Inter-domain Routeing Information Exchange Protocol, 1990.

[5] Lougheed, K. and Y. Rekhter, "A Border Gateway Protocol (BGP)," RFC 1105, June, 1989.

[6] Lougheed, K. and Y. Rekhter, "A Border Gateway Protocol (BGP)," RFC 1163, June, 1990.

[7] Mills, D., "Exterior Gateway Protocol Formal Specification," RFC 904, April, 1984.

[8] Rekhter, J., "The NSFNET Backbone SPF based Interior Gateway Protocol," RFC 1074, October, 1988.

[9] Rekhter, J., "EGP and Policy Based Routing in the New NSFNET Backbone," RFC 1092, February, 1989.

[10] Willis, S. and J. Burruss, "Experimental Definitions of Managed Objects for the Border Gateway Protocol (Version 2)," Internet Draft "`draft-ietf-iwg-bgp-mib-01.txt`," September, 1990.

YAKOV REKHTER received his M. S. in Physics from Leningrad University (USSR) and his M.S. in Computer Science from New York University. Yakov is a Research Staff Member at the T. J. Watson Research Center of IBM Corporation. He is a chairman of the BGP Working Group of the IETF and an active participant in the ANSI X3S3.3. For the last 4 years his major focus was on the NSFNET Backbone routing, network management, and other NSFNET Backbone related projects.

DAVE KATZ is an Internet Engineer on the NSFNET Backbone project for the Merit Computer Network. He is active in standards activities in both the DoD Internet (the Internet Engineering Task Force) and OSI (ANSI X3S3.3, OSI Network/Transport layers). Dave is an agnostic when it comes to network religion.

Note

[Ed.: Both the Interconnectivity Working Group (IWG) and the Open Routing Working Group (ORWG) have officially changed names since the articles in this issue were written. The IWG is now called the *Border Gateway Protocol Working Group,* and the ORWG is now called the *Inter-domain Policy Routing Working Group.* The new names were chosen to better reflect the charters of these groups].

RFC 1174: Federal Networking Council Drops "Sponsoring Agency" Requirement For Internet Network Name and Number Databases

by Daniel P. Dern

Background

At the July 1990 meeting of the Internet Engineering Task Force (IETF) held in Vancouver, Canada, it was announced that the Federal Networking Council (FNC), an interagency forum that oversees the operation and evolution of the Internet, has accepted a proposal submitted by the Internet Activities Board (IAB) to revise the way new organizations "join the Internet"—obtain and register their numeric and name ID data in the Internet databases, actions which are essential to connectivity and being reachable.

The Internet is the world-wide set of interconnected networks and organizations running the IP (Internetworking Protocol) protocol suite which accreted around, and has subsumed, the original ARPANET. Membership and access to the Internet has historically been guided by regulations which reflected the needs of Federal agency backbone networks.

Changing times

But these rules don't always make sense in today's Internet. Increasingly, corporations are connecting to the new commercial IP networks such as Performance Systems International's *PSInet* (Reston, VA) and Uunet's *Alternet* (Falls Church, VA), as well as to "regional" networks launched under the NSFNET initiative. At the same time, international organizations, for whom these historic restrictions and procedures are equally inapplicable, are seeking Internet access. Yet even for users not looking for access to restricted backbones, getting network identifiers is essential—and as a more diverse community is created, new access controls are needed.

Richer community

The IAB proposal will remove what is universally acknowledged to be an obsolete bureaucratic impediment to a global network which now serves corporate and international members as well as its original community of government, research and academic organizations. "We have asked the IAB to proceed with the necessary procedures to implement their recommendations," reports Bill Bostwick, executive director for the Federal Networking Council. "The FNC's viewpoint is this will create a much richer community of users, and let individual networks each manage their appropriate use policies."

"This is marvelous news for the internetworking community," comments Dan Lynch, President of Interop, Inc. "It will spur even more rapid Internet growth—organizations won't hold back because they feel that 'have to qualify,' and will have more choice of which network to join."

RFC 1174

The IAB proposal was submitted by Dr. Vinton Cerf, Vice President of the Corporation for National Research Initiatives (NRI) and Chairman of the (IAB), in the form of RFC 1174.

Specific procedural changes recommended in RFC 1174 include:

- Dropping the concept of "connected status" in favor of an "Intended Usage" database
- Eliminating the "sponsoring agency" requirements for entry of organization's network IDs into Internet network databases
- Sub-delegating assignment of network ID numbers to selected international and other organizations.

Appropriate use

"Appropriate use" is a major concern within the evolving Internet community. Mission-specific backbones networks belonging to federal agencies, such as the National Science Foundation *NSFNET* and NASA's *NASA Science Internet,* have explicit statements as to what organizations may send traffic over them. Commercial and regional networks often have different usage policies.

Traffic that stays within a given network is not subject to usage rules established for other backbones. The challenge arises for "transit traffic"—packets that flow from one network to or across another, such as between NSFNET and BARRNet, the Bay Area Regional Research Network. "We are primarily interested in identifying whether the traffic would fit the 'acceptable use' policies for the various national agency backbone networks," says Cerf.

But to join a regional or commercial IP Internet network, an organization must obtain several essential Network Identifiers, and have this information entered in the Internet Registry databases—even if the organization doesn't want to access these restricted backbones. The IAB proposal's intent is to simplify this process, and bring it in line with the changing nature of the Internet.

Hans-Werner Braun, Principal Investigator on the NSFNET backbone project, says, "The new procedures will streamline international participation in the Internet community—which is growing by leaps and bounds."

Registration

The Internet databases at issue are managed and distributed by the *DDN Network Information Center* (DDN NIC), run by SRI International in Menlo Park, CA, under contract to the Defense Communications Agency (DCA). The SRI NIC derives its registration authority from the *Internet Assigned Numbers Authority* (IANA) at USC Information Sciences Institute (USC-ISI), Marina del Rey, CA. Global uniqueness of these numbers is essential to operation of the Internet.

The Internet databases managed by the DDN NIC include *IP Network numbers,* used in IP network addresses; *Autonomous System Numbers* (ASN), used in Internet routing; and *domain names.* The *Domain Name Server* (DNS) database is the mechanism by which name strings in text form, are translated into an IP address, e.g., translating the host/domain name "nri.reston.va.us" to IP address "132.151.1.1".

"Your network needs to be listed in the Domain Server, or else it's hard for people to find you via your IP address," explains Vint Cerf. "Normally, you ask your Domain Name Server to translate a string-name, such as "host.domain," into an IP address. And without an address you can't open a connection or send an IP datagram to the target—e.g., do remote login or file transfer. So you need to be in that database to be a full operating net in the Internet."

The IAB recommendation would let all applicants receive numbers in the IP Network Number and Autonomous System Number databases, and allow any registered network to be entered into the DNS database, without regard to connected status.

Usage statements

RFC 1174 recommends new mechanisms to implement Internet access and traffic restrictions. One is the creation of a NIC database of "Usage Statements" to be submitted by all members of Internet networks, regarding what types of traffic (commercial, research, academic, administrative, etc.) and volume levels they anticipate sending outside their immediate network connection.

continued on next page

RFC 1174 *(continued)*

Policy routing

"This statement can be short, but its intent is to allow a database to be built which will associate a given host/net ID with a type of traffic expected to be seen from that source," explains Cerf. "In particular, if there are networks with usage constraints, such as not wanting to carry arbitrary commercial traffic, they might choose not to accept or provide routes to and from such networks that don't fall into their 'acceptable use' envelope."

Each member Internet member would be able to specify a list of network numbers to be excluded from their routing tables during route set-up, which is when the routing tables determine what networks are reachable, and what to use to reach them.

"If there is no table entry announcing that a given network is available, traffic can't get there," says Cerf, "All the leading vendor products used in the Internet, and certainly all the NSF 'regional' networks, can implement these filters."

Steven Goldstein, NSF Program Director for Interagency and International Networking Coordination, adds, "The policy information in the Usage database makes this type of policy-based routing practical."

The next step in policy-based routing is development of a mechanism that can examine, route and filter individual packets selectively, on a real-time basis—for example, to permit electronic mail but deny remote login from a given network, or select an appropriate backbone based from several connected choices. Opening up the NIC databases and dropping "sponsorship" requirements, while generally hailed as a major step forward in the growth of the Internet, should not be considered the final step, all agree.

"The fact that major backbones have any traffic restrictions implies that users may encounter unanticipated barriers to connectivity, even for e-mail," comments Martin Schoffstall, Vice President of Performance Systems International, whose PSInet is one of the international commercial IP networks. "And many organizations are understandably leery of important traffic possibly—and unpredictably—flowing through networks belonging to their competitors."

Along with network- and packet-level policy-based routing mechanisms, unrestricted, "politically neutral" commercial backbones are seen as one possible solution to the quandary. But for now, all agree that "opening the databases" will be a welcome, landmark event.

Good news

"The IAB proposal is good news—it recognizes that the Internet isn't 'just a big ARPANET' anymore," says Dr. Stephen Wolff, the Director at the NSF's Division of Networking and Communications Research and Infrastructure (DNCRI).

"This reflects the reality that the Internet is a world wide event," Interop's Lynch concludes. "It confirms that IP networking is an important part of how we work."

References

[1] "IAB Recommended Policy on Distributing Internet Identifier Assignment and IAB Recommended Policy Change to Internet 'Connected' Status," RFC 1174.

[2] B. Kahin, Editor "Commercialization of the Internet, Summary Report," RFC 1192.

DANIEL P. DERN is a Watertown, Mass-based free-lance writer specializing in technology, science and industry. A frequent contributor to *ConneXions,* including last year's ARPANET historical retrospective, Dern writes for leading publications and vendors in the network and computer industry. He can be reached via e-mail as `ddern@world.std.com` (Internet) or `dandern` (MCIMail).

Book Review

Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architectures, Second Edition, by Douglas E. Comer, Prentice-Hall, 1990, ISBN 0-13-468505-9.

Single reference work

The Internet suite of Protocols is rich collection of standards, specifications, and oral tradition which together form a magnificent basis for modern computer-communications. Until 1987, there was no single reference which could provide both overview and detailed explanation of all the bits and pieces. That watershed was the publication of Doug Comer's "Internetworking with TCP/IP." This book has no peer in the annals of the written tradition of the Internet suite—no one document enjoys the breadth and depth of topics covered. In mid-1990, the Second Edition of this tome was published.

New topics

Comer's Second Edition has all the good qualities for the first: a blend of architecture and protocol spanning the core elements of the protocol suite. Further, the 2nd Ed. is some 43% larger, as Comer extends the core which he explains in a crisp, academic style. The inclusion of new topics partially addresses the major, inevitable, weakness of the first edition—the Internet suite is simply too large to be fully explained in any one work. In the 2nd Edition, new topics include:

- OSPF (the *Open Shortest-Path-First* routing protocol)
- Multicasting in the Internet suite with IGMP
- BOOTP (diskless booting protocol)
- Terminal Service (primarily TELNET)
- File Service (primarily FTP and TFTP)
- Mail Service
- Network Management

In brief, although some emphasis is placed on technology recently consolidated, the main additions are in the core areas omitted in the first edition: *application services.*

Of course, as with any large work, there are some complaints: for example, the treatment of the historical *GGP* (gateway-to-gateway protocol) is largely unnecessary. It could be replaced by a more thorough treatment of the new *Border Gateway Protocol* (BGP).

Basic coverage

But, my fundamental criticism of the 2nd edition is that Comer still does not extend far enough in his treatment: in each of the application areas, the technology covered is very basic. For example, the treatment of the Berkeley-specific network services (*rlogin, rsh,* et. al.) is still quite small. Although these services are not "de jure" components of the Internet suite, they are, nonetheless important "de facto" parts as evidenced by the dominance of systems which implement these extensions.

Perhaps this criticism is unfair given the possible scope of the topic. However, a key aspect of the Internet suite is that it has dozens of excellent services. By limiting discussion to only the "inner circle" of protocols, a reader may not fully appreciate the richness and diversity of the protocol suite.

Buy this book!

Regardless of one's opinion in this matter, it is clear that the second edition is a worthy successor. Whether you own the first edition or not, you should buy the second: it is without peer. —*Marshall T. Rose*

[Ed.: A companion book, *Internetworking with TCP/IP Volume II: Implementation and Internals,* by Douglas Comer and David Stevens, has also been published recently. It will be reviewed in a future issue].

Upcoming Events

24 courses

Interop, Inc. is pleased to announce the *1991 Internetworking Tutorials* program. These 24 courses are taught by leading experts in the networking field, and will be offered at 3 separate locations in the US. (Not all tutorials are offered at every location, call us at 1-800-INTEROP or 415-941-3399 for more information). The tutorials are:

Introduction to the TCP/IP Protocol Suite	Dr. Douglas Comer
TCP/IP: Internals and Implementation	Dr. Douglas Comer
The Internet Approach to Multivendor Enterprise Internets	James Herman & Peter Sevcik
How to GOSIP™	Richard desJardins
Understanding OSI	David M. Piscitello
Practical Perspectives on OSI Networking	Dr. Marshall T. Rose
ISODE Internals	Dr. Marshall T. Rose
The X.400 Message Handling Systems: Standards & Practice	Richard desJardins
The X.500 Standards for Directory Services	Dr. Sara Radicati
Distributed File Systems and NFS™	Dr. Ralph Droms
Introduction to the X Window System	Dr. Wayne R. Dyksen
Introduction to the X Toolkits	Dr. John T. Korb
The Simple Network Management Protocol (SNMP) for Internet Network Management	Dr. Jeffrey D. Case
Application of Bridges and Routers: Network Design & Product Survey	Scott Bradner
Theory of Bridges & Routers: Protocols & Algorithms in Depth	Dr. Radia Perlman
Network Security— The Kerberos Approach	Jon A. Rochlis & Jeffrey I. Schiller
Distributed Computing— Project Athena, A Working Example	Jon A. Rochlis & Jeffrey I. Schiller
Network Administration & Security for UNIX-based Networks	T. Eric Brunner & Ronald Natalie
Internet Naming and Directory Services	Brendan Reilly
LAN Interconnection Architectures: Solutions, Tradeoffs, and Trends	Dr. Gilbert Falk & Tony Michel
A Practical Introduction to Network Performance	Dr. David D. Clark
Gigabit Network Architectures	Craig Partridge
IBM Systems Network Architecture (SNA) Interoperability	Dr. Wayne Clark
UNIX Network Programming	Dr. Richard Stevens

Dates and locations

March 18–21, 1991:	*Washington DC* (Hyatt Regency, Crystal City)
April 22–25, 1991:	*Boston* (Lafayette Hotel)
May 13–16, 1991:	*Dallas* (Grand Kempinski Hotel)

Call For Papers

ACM SIGCOMM '91: Communications Applications, Architectures and Protocols, will be held in, Zurich, Switzerland, September 4–6, 1991 with Tutorials on September 3, 1991. The conference provides an international forum for the presentation and discussion of communication network applications and technologies, as well as recent advances and proposals on communication architectures, protocols, algorithms, and performance models. It is the first time that SIGCOMM will hold its conference outside of the United States.

Topics

Authors are invited to submit full papers concerned with both theory and practice. The areas of interest for the conference include, but are not limited to the following: Analysis/design of computer network architectures/algorithms, innovative results in local area networks, computer-supported cooperative work, network interconnection and mixed-media networks, high-speed networks, resource sharing in distributed systems, network management, distributed operating systems and databases, protocol specification, verification, and analysis.

Papers

Papers should be about 20 double-spaced pages long and should have an abstract of 100–150 words. All submitted papers will be reviewed and will be judged with respect to their quality and relevance. The Proceedings will be distributed at the conference and published as a special issue of ACM SIGCOMM *Computer Communication Review.* Notable papers will be considered for publication in ACM *Transactions on Computer Systems.* Submit 5 copies of each paper to:

Prof. Bernhard Plattner, Program Chairman
Technische Informatik und Kommunikationsnetze
ETH-Zentrum (IFW)
8092 Zurich, Switzerland
Telephone: +41 1 254-7000 Fax: +41 1 262-3973
E-mail: plattner@komsys.tik.ethz.ch

Greg Wetzel, US Program Committee Contact
AT&T Bell Laboratories
2000 N. Naperville Road, M/S IH 1B-213
Naperville, IL 60566
Telephone: +1 (708) 979-4782 Fax: +1 (708) 979-2350
E-mail: gfw@pueblo.att.com

Special Sessions

One or more sessions of the conference will be devoted to the subject of applications of high speed networks. Papers in these sessions will focus on concepts, implementation and experience of applications that need the performance that future high speed networks will offer. Topics include, but are not limited to new approaches to computer-supported cooperative work, graphic visualization, and access to supercomputers. Papers submitted for these topics should address applications and their communications requirements.

Student Paper Award

Papers submitted by students will enter a student-paper award contest. Among the accepted papers, a maximum of four outstanding papers will be awarded (1) full conference registration and (2) a travel grant of $500 US dollars. To be eligible the student must be the sole author or a primary contributor to the paper.

Important dates

Deadline for paper submission:	March 2, 1991.
Notification of acceptance:	April 30, 1991.
Camera ready papers due:	May 31, 1991.

Contact sicomm91@nri.reston.va.us for more information.

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

ADDRESS CORRECTION
REQUESTED

CONNEXIONS

EDITOR and PUBLISHER — Ole J. Jacobsen